Self-Organisation in Traffic Signal Control Algorithms

Samantha Jane Movius

UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

Dissertation presented for the degree of
Doctor of Philosophy
in the Faculty of Engineering at Stellenbosch University

Promoter: Prof JH van Vuuren

March 2018

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 1, 2018

Abstract

Two popular types of traffic signal control are fixed-time control and vehicle-actuated control. The latter method involves switching traffic signals based on detected traffic flows and thus offers more flexibility than the former, which relies solely on cyclic, predetermined signal phases. The notion of self-organisation has relatively recently been proposed as an alternative approach towards improving traffic signal control, particularly during periods of light traffic flow, due to its flexible nature and its potential to result in emergent behaviour.

The effectiveness of five self-organising traffic signal control strategies from the literature, as well as a fixed-time control strategy, have previously been compared in a simulated environment. Various shortcomings of three of these algorithms are pointed out in this dissertation and algorithmic improvements are suggested to remedy these deficiencies. The significant improvements resulting from these algorithmic modifications are then quantified by means of their implementation in a newly designed agent-based, microscopic traffic simulation model.

Two novel self-organising traffic control algorithms are also proposed in this dissertation. These algorithms have been designed in such a way as to avoid certain shortcomings discovered in the aforementioned algorithms. The two novel algorithms, together with the improved versions of the three existing algorithms and the remaining pair of algorithms from the literature, are also subjected to thorough testing in the aforementioned simulation framework in terms of their propensity to facilitate the formation of green waves and to recover from various disruptions (such as road closures or abnormal traffic induced by large events) within the context of both gridded street networks and corridors with approaching side roads. All eight algorithms are finally implemented in a simulation model representing an existing road network in order to compare and evaluate the effectiveness of these algorithms within the context of a real-world scenario.

It is found that the two newly proposed algorithms outperform existing self-organising traffic signal control algorithms under certain traffic conditions and road network topologies.

Uittreksel

Twee populêre tipes verkeersbeheertegnieke is vaste-tyd beheer en voertuig-geïnduseerde beheer. Volgens laasgenoemde tegniek berus die wisseling van verkeerseine op die meting van verkeersvloei wat meer buigsaamheid bied as die eersgenoemde tegniek wat berus op sikliese, voorafbepaalde seinfases. Die begrip van self-organisering is relatief onlangs as 'n alternatief vir verbeterde verkeerseinbeheer voorgestel as gevolg van die buigsame aard en die potensiaal daarvan om te lei na ontluikende gedrag.

Die doeltreffendheid van vyf self-organiserende verkeersbeheerstrategieë uit die literatuur is reeds voorheen in 'n gesimuleerde omgewing vergelyk. Verskeie tekortkominge van drie van hierdie algoritmes word in hierdie proefskrif uitgewys en algoritmiese verbeterings word voorgestel om hierdie tekortkominge uit die weg te ruim. Die beduidende verbeterings wat uit hierdie algoritmiese veranderinge voortspruit, word dan aan die hand van 'n nuut-ontwerpte agent-gebaseerde verkeers-mikrosimulasiemodel gekwantifiseer.

Twee nuwe self-organiserende verkeersbeheeralgoritmes word ook in hierdie proefskrif daargestel. Hierdie algoritmes is ontwerp om sommige tekortkominge wat in die bogenoemde algoritmes ontdek is, aan te spreek. Die twee nuwe algoritmes, tesame met die verbeterde weergawes van die drie bestaande algoritmes sowel as die oorblywende paar algoritmes, word ook aan deeglike toetse in die bogenoemde simulasieraamwerk onderwerp in terme van hul vermoë om die vorming van groen golwe te bewerkstellig en om van ontwrigtings (wat deur straatsluitings of abnormale verkeersdruk as gevolg van groot byeenkomste te weeg gebring word) in 'n netwerk van straatblokke en in 'n korridor met aansluitende systrate te herstel. Al agt algoritmes word laastens ook in 'n simulasiemodel van 'n bestaande padnetwerk geïmplementeer om sodoende die doeltreffendheid van die algoritmes in die konteks van 'n realistiese scenario te vergelyk.

Daar word in al die simulasie-eksperimente bevind dat die twee nuwe algoritmes die bestaande self-organiserende algoritmes onder ligte verkeerstoestande uitstof, terwyl 'n eenvoudige vaste seinfase-siklus beheerstrategie die beste benadering onder swaar verkeerstoestande is.

Acknowledgements

The author wishes to acknowledge the following people and institutions for their various contributions towards the completion of this work:

- My supervisor, Prof JH van Vuuren for all the effort and time put into aiding me in my research. I feel very privileged to have had a supervisor who takes such an interest in my work academically, as well as in me as a person. Thank you, Prof, for taking me on as your student even though I was from another department — I feel like this was exactly where I was suppose to end up. “Samamfa” will definitely keep in touch.
- My fellow SUnORE members, for the interesting tea time chats and coffee breaks. Without you guys, this experience would not have been the same
- Almost-Dr Anthony Smith, for helping me count cars for an entire day. You’re not so bad after all.
- The Industrial Engineering Department at Stellenbosch University as well as SUnORE, for the office space and computing facilities.
- Prof Martin Kidd, for all the help with the statistical analysis of the data.
- SUnORE, the National Research Foundation and the Harry Crossley Foundation are acknowledged with gratitude for the financial support received over the past three years.
- The Anylogic support team, and in particular Gregory Monakhov, who responded to all my emails over the last year. The timely and extremely helpful responses are greatly appreciated.
- My parents, Michael and Stephanie, for all the support and encouragement they have given me over the entire course of my studies. A special thank you to my father who dedicated many hours of his time trying to help me with Java, even when I was being “otherwise.” I love you both.

Table of Contents

List of Acronyms

Acronym	Meaning
AAC	Anticipated All Clearing
ANOVA	Analysis Of Variance
AT	Adam Tas
CONTRAM	CONtinuous TRaffic Assignment Model
EOQ	Economic Order Quantity
I-TSCA	Inventory Traffic Signal Control Algorithm
IRC	Interactive Run Controller
IUMSM	Intersection Utilisation Maximisation Supervisory Mechanism
LSD	Least Significant Difference
LWR	Lighthill-Whitham-Richards
O-TSCA	Osmosis Traffic Signal Control Algorithm
PBE	Platoon-Based Extension
PBS	Platoon-Based Squeezing
PBSS	Platoon-Based Self Scheduling
PMI	Performance Measure Indicator
SCATS	Sydney Coordinated Adaptive Traffic System
SCOOT	Split Cycle Offset Optimisation Technique
SOTL	Self-Organising Traffic Lights
SR-TSCA	Saturation Ratio Traffic Signal Control Algorithm
SUMO	Simulation of Urban MObility
TRANSYT	TRAffic Network StudY Tool
VP-TSCA	Vehicle Platoon Traffic Signal Control Algorithm

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Contents

1.1 Background

Traffic congestion is a major concern in large cities all over the world [20, 51, 78]. This is due to the economic, environmental and social consequences that are commonly associated with heavily congested roads. The main cause of traffic congestion is the over-utilisation of roads which has the potential to lead to dense, stop-and-go traffic [24]. Roadway widening construction is a popular way of obtaining temporary relief from heavily congested roads. This solution is, however, unsustainable and only offers short term success as a result of the ever-growing global population [17]. Over time, the improved roads once again become congested and so the cycle continues.

One of the longest traffic jams ever recorded (shown in Figure 1.1) occurred in August 2010 on the China National Highway 110, and lasted over ten days [84]. Vehicle queues spanned a distance of over 100 km, and many drivers trapped in these queues were only able to travel 1 km per day, and some reported being stuck in the traffic jam for up to five days. While this scale of traffic congestion is less common, the reality is that traffic congestion is an issue that will continue to grow until sustainable preventative measures are taken [56].

A viable alternative for reducing urban traffic congestion involves the optimisation of traffic signal control algorithms employed at signalised intersections. Improved traffic signal control may serve to dilute concentrated traffic in road networks by increasing the efficiency of signal duration times, leading to reduced vehicle delay and promoting the formation of so-called *green waves*[1].

There are two distinct types of traffic signal control: *fixed time control* and *vehicle actuated control* [26]. Fixed time control involves the specification of predetermined signal cycle times based on expected traffic flow densities for various time periods. This method of control may

[1]Green waves refer to traffic flow that propagates uninterrupted through a number of consecutive intersections.

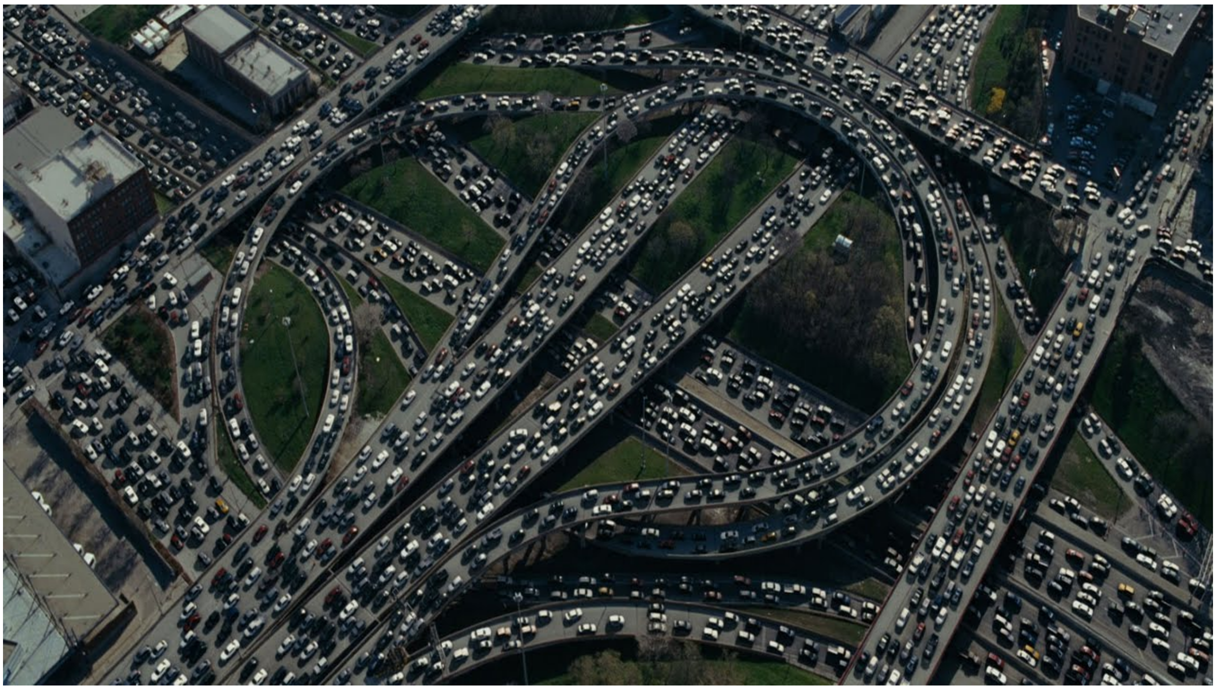


FIGURE 1.1: *One of the longest recorded traffic jams in history [84].*

be coordinated by allowing a number of successive intersections to have the same cycle length, together with a pre-timed offset[2]. While this is not problematic to implement in a one-way street, it becomes more complex when attempting coordination in two directions, and even more so when coordinating three or more directions. The *bandwidth* is the length of the time interval in which vehicles are able to travel at the progression speed without receiving a red signal, and is shown in Figure 1.2 along with the cycle time and signal offset. A disadvantage of fixed control, however, is that traffic volumes often fluctuate dramatically over the course of a single day such that actual demand is typically not met by such predetermined cycle times and, as a result, green times are often either too long or too short [12].

Vehicle actuated control, on the other hand, is capable, at least to some extent, of adapting according to real-time traffic conditions of the road network. Unlike fixed time control, vehicle actuated control is responsive to changes in traffic flow, but requires the implementation of vehicle detection equipment in order to register the prevailing traffic conditions. A basic form of vehicle actuated control involves awarding a specific phase of green time once a vehicle has been detected at the intersection. An example of this is shown in Figure 1.3. A phase receives green time until a *critical gap-out*[3] occurs, assuming that the green time elapsed is above a minimum green time value τ_{min} and less than a maximum green time value τ_{max} [12].

A widely used form of vehicle detection is the inductive-loop traffic detector, whereby a wire sensor loop is installed in the road pavement just before an intersection [58]. Vehicles that pass over the sensor or stop within the loop of the sensor, induce a current in the wire, decreasing its inductance. If the inductance falls below a certain threshold, the loop detector sends a signal to the traffic signal controller that is responsible for allocating green time to different directions within the intersection. This signal triggers a switching of the traffic signals. A disadvantage of such a detection system is that the system cannot measure the velocity of an incoming vehicle.

[2]Offset refers to the difference in time between the start of a green signal at two neighbouring intersections.

[3]A critical gap-out is a threshold during which no new vehicles approach the intersection. The corresponding time of this gap-out is referred to as the gap-out time.

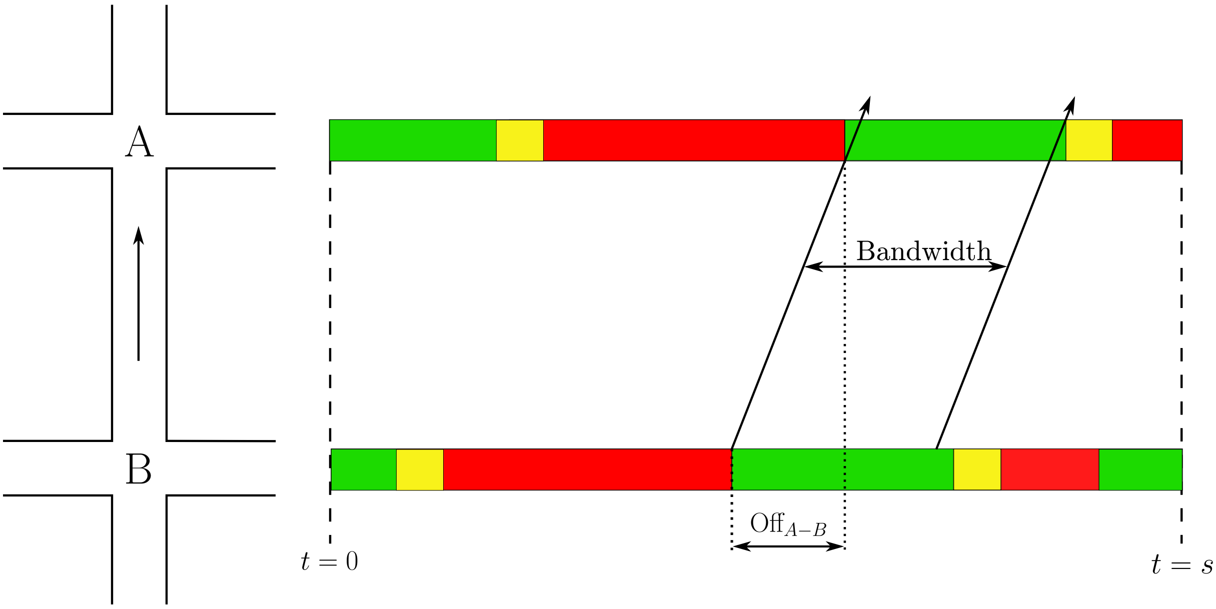


FIGURE 1.2: *The coordination of traffic signalling at two intersections A and B along a one-way street. The offset is represented by* Off_{A-B} *and the length of the cycle time s.*

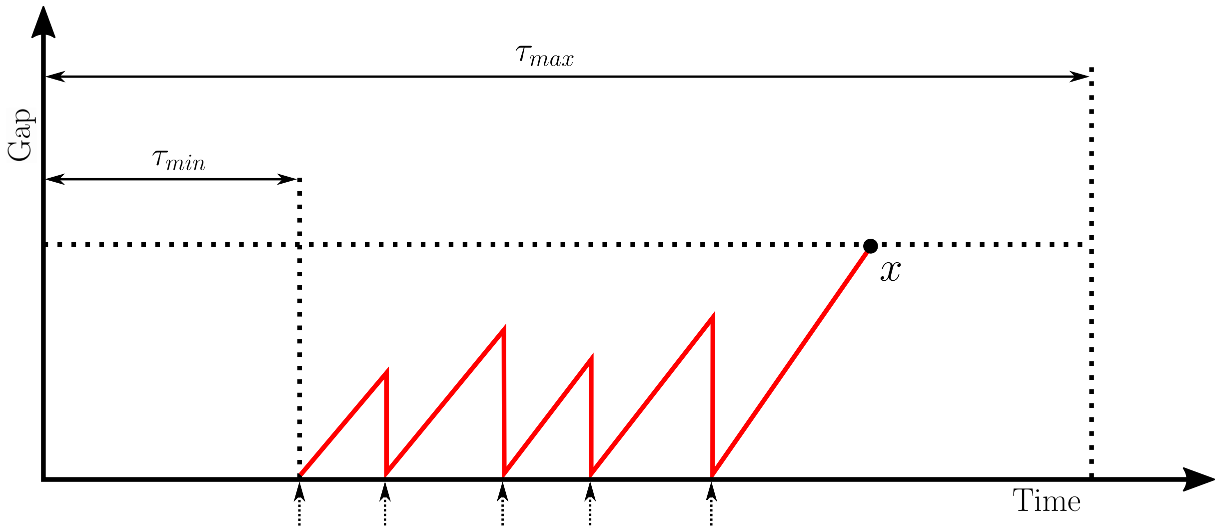


FIGURE 1.3: *An example of basic vehicle actuation. The vertical arrows along the horizontal axis represent actuation occurring as a result of incoming vehicles, while x is the time at which there is a gap-out. Here* τ_{min} *and* τ_{max} *are the minimum and maximum green times, respectively.*

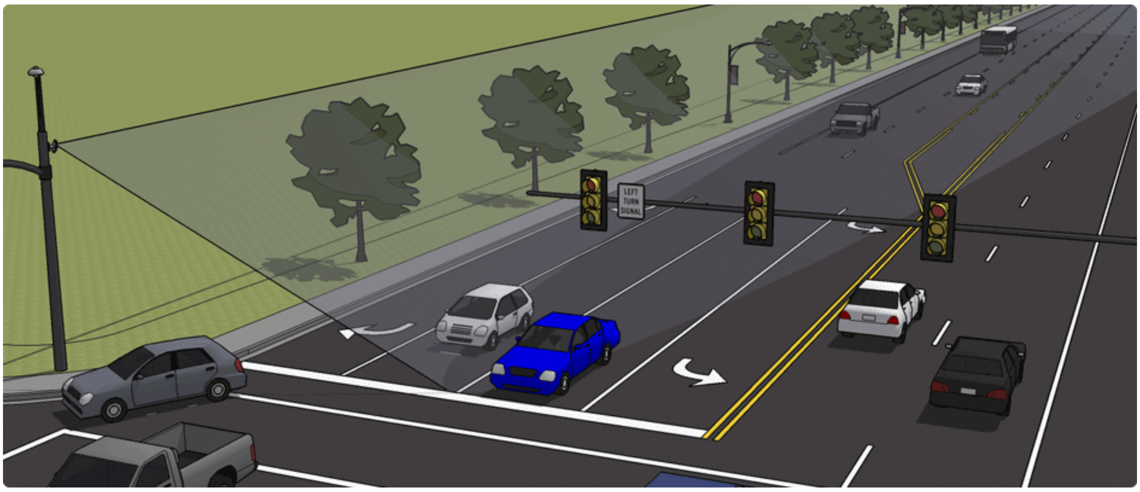


FIGURE 1.4: *The SmartSensor Advance Extended Range radar detection unit mounted at an intersection and its detection region [82].*

The detection sensor also lies within the road surface, which may cause traffic disruptions when installing and maintaining these systems [41].

A less intrusive and more advanced detection mechanism is that of radar detection, such as afforded by the *SmartSensor Advance Extended Range* detection unit, illustrated in Figure 1.4 [82]. The detector is mounted 5 to 12 metres up a vertical mast arm at an intersection and is able to detect approaching vehicles from up to 275 metres away. Once a vehicle is detected by the sensor, individual characteristics of the vehicle may be determined, including the vehicle speed, range and estimated arrival time at the intersection [82]. It is also capable of detecting the time, location and size of gaps in traffic, which are useful data that are communicated to the signal controllers. An advantage of the radar sensor over the inductive loop detector is that the sensor is installed above ground at an intersection and therefore no roadway construction is necessary for its installation, maintenance or replacement. The radar sensor is also more resistant to poor weather conditions, including freezing rain, snow, wind, dust and fog, than other forms of detection involving image processing. Due to all these advantages, the *SmartSensor Advance Extended Range* detection unit is assumed to be the mode of vehicle detection in this dissertation.

Two common vehicle actuated traffic control strategies in use today are the *Split Cycle Offset Optimisation Technique* (SCOOT) and the *Sydney Coordinated Adaptive Traffic System (SCATS)* [52]. While these two controls are adaptive and are able to adjust traffic flow cycle times so as to benefit current traffic conditions, these systems are centralised. A disadvantage of centralised systems is that the global optimisation of signal times across all intersections is NP-hard and thus an optimal solution often cannot be found in real-time for large traffic networks [46, 60, 77]. The use of a decentralised traffic control system is advantageous as the problem of traffic control at each intersection may be viewed as an isolated problem, requiring no explicit information on how signals are controlled at neighbouring intersections, thus significantly reducing the complexity of decisions related to signalised intersection traffic control.

The decentralised paradigm of *self-organisation* has been suggested as an appropriate approach toward developing effective traffic signal control algorithms. In order for a process to exhibit self-organising behaviour according to [15], a number of properties must be present in the system. First, an increase in order must be exhibited [15, 70]. This refers to improved organisation of the system as a whole, when specific parts of the system are ordered to increase the performance of a certain function [30]. Secondly, the system must be free from all kinds external control [15, 30, 69]. Input is allowed into the system, so long as it does not affect future decisions related to the system. Thirdly, the system should exhibit adaptable behaviour to changes, and be able to maintain the organisation it had before the changes [15, 30]. Finally, the system has to be dynamic as self-organisation is a process that occurs over time [15, 30, 70]. If a process exhibits each of these key characteristics, then it is self-organising. Self-organisation is a good approach towards signal control not only because it is decentralised, but also due to the fact that it has the potential to lead to the natural *emergence* of coordination between intersections.

The term *emergence* is often confused with self-organisation in literature. In order to avoid this confusion, specific properties are emphasised in [15] that must be present in order for emergent behaviour to be possible. The first property is that the behaviour of the system at the macro-level must stem directly from interactions between individual parts at the micro-level [15, 70]. The second property is referred to as *radical novelty*. It requires that the system is capable of exhibiting novel behaviour, while the elements in the macro-level are irreducible to the elements of the system at the micro-level [15, 29, 57, 70]. Coherence is the third property and refers to the presence of an unwavering correlation between parts of the system [15, 57]. The next property states that interactions need to occur between parts of the system in order to ensure a novel behavioural outcome. The fifth property requires that the system be dynamic, as emergence typically only occurs after a certain point in time [15, 57]. The next property is decentralised control, meaning that there is no single central controlling unit that is responsible for the system as a whole [15, 57]; only local control may be present, which does not directly control the system as a whole. The seventh property is known as a two-way link. This refers to the continuous influence that the micro-level elements and the macro-level elements have on one another [15, 57]. The last property that should be present in an emergent process, is flexibility, requiring that the system should be able to recover quickly from any disturbances [15, 30].

Self-organisation can exist without leading to emergence and emergence can occur without the prerequisite of self-organisation, while self-organisation can also lead to emergence, and *vice-versa* [15, 29, 69, 70]. In [16] it is claimed that self-organisation in traffic signal control can lead to emergence of favourable coordination in intersections. A self-organising traffic signal control algorithm proposed by Lämmer and Helbing [46], a self-organising traffic signal control algorithm developed by Gershenson and Rosenblueth [22], and three self-organising traffic signal control algorithms recently proposed by Einhorn [16] appear to be effective in terms of being capable of reducing vehicle delay time in road networks. The aforementioned algorithms require the use of radar vehicle detection equipment mounted at each intersection and most of them require predetermined parameter values.

1.2 Informal problem description

As mentioned in the previous section, self-organisation is a promising approach in respect of improving the adaptive signal control methods that are currently in use. In particular, the aforementioned five self-organising algorithms by Gershenson and Rosenblueth [22], Lämmer and Helbing [46], and Einhorn [16] appear to be effective in terms of reducing vehicle delay as well as facilitating the propagation of green waves through successive intersections. These five self-organising algorithms have previously been compared to one another as well as to a fixed-time control algorithm. This comparison took place in a rather simplistic, simulated environment using a number of *performance measure indicators* (PMIs) to determine which algorithms were most effective. The results revealed that in a four-intersection corridor road network, the hybrid algorithm outperformed the others overall, while in a 3×4 grid of intersections, the osmosis-inspired algorithm was the most effective [16]. These results were, however, obtained under the assumption that neighbouring intersections (one of which is depicted in Figure 1.5) lie at equal distances from one another, which is an unlikely occurrence in practice. Einhorn [16] also did not consider road networks of different sizes or include the presence of disruptions such as road closures, which are an everyday reality. It may be that the results obtained by Einhorn [16] are only valid for the specific scenarios considered in his dissertation. One of the research aims in this dissertation is therefore to ascertain whether similar results are obtained in other road network configurations, adopting an independent and newly developed simulation model as a test bed. Secondly, the aforementioned algorithms are scrutinised in order to ascertain whether any alterations to their working will be beneficial in respect of their performance. Lastly, the relative performance of these algorithms are assessed in the context of a variety of test scenarios.

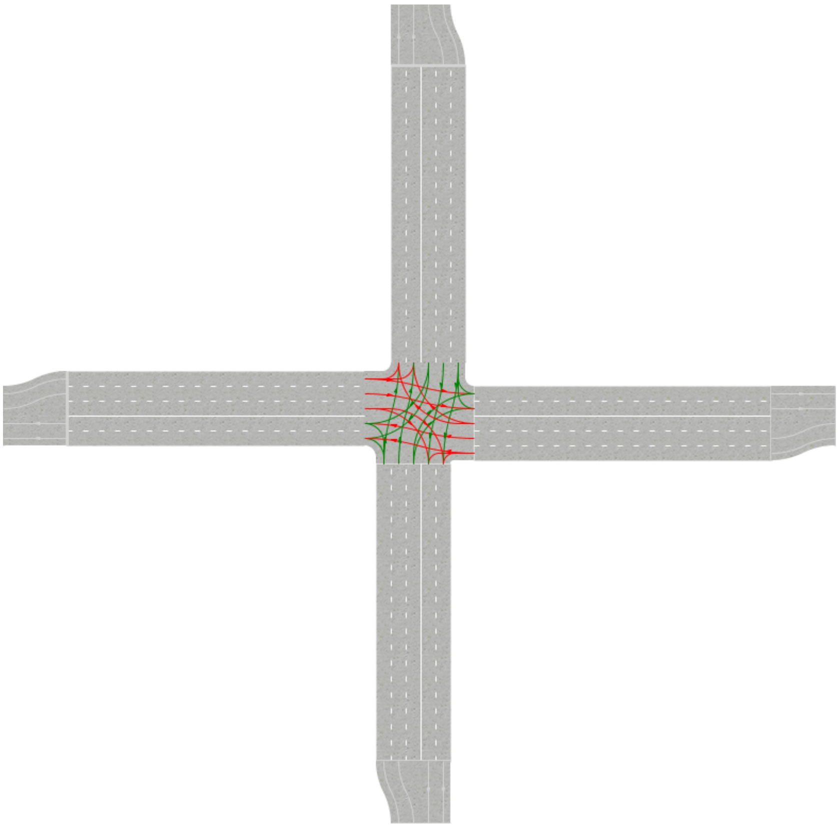


FIGURE 1.5: *An image of an intersection taken from the simulation created in this dissertation. There are three approach lanes from each direction. Vehicles in the left lane can either turn left or continue straight through the intersection, vehicles in the middle lane can only travel straight through the intersection, while vehicles in the right lane may only turn right.*

The overarching research goal in this dissertation is to propose new self-organising algorithms and to compare their relative performances with those of existing self-organising algorithms over a range of traffic density scenarios, as well as to measure their relative performance within various transportation network topologies. This is achieved by building a microscopic traffic simulation model anew and adopting the PMIs proposed by Einhorn [16] together with two additional PMIs in order to determine whether there is a significant change in the relative performance of the algorithms when enhancing the original simulation model's realism by including unequal spacings between intersections, increasing road corridor sizes, incorporating road closures and accommodating varying vehicle arrival rates. Furthermore, the algorithms are compared in a model of a real-world transportation network, employing real, recorded traffic flow rates for this road network model.

1.3 Scope and objectives

The following ten objectives are pursued in this dissertation:

I To *conduct* a thorough survey of the literature related to:

(a) the notion of self-organisation and emergence, and how they may be applied to traffic control,

(b) traffic signal control algorithms based on the paradigm of self-organisation,

(c) simulation modelling techniques, with a focus on microscopic traffic simulation modelling.

II To *create* a new microscopic traffic simulation model for use in the context of urban road networks. This model should be able to incorporate the self-organising traffic signal control algorithms researched in pursuit of Objective I(b) above and should be informed by the research conducted in pursuit of Objective I(c) above.

III To *validate* the traffic simulation model of Objective II according to standard model validation principles and guidelines.

IV To statistically *compare* the relative effectiveness of a number of existing self-organising algorithms by embedding them in the simulation model of Objectives II and III within the context of various road network traffic conditions and topologies of different complexities. These complexities should include:

(a) accounting for unavoidable disturbances in the road network, such as road construction which may result in a road closure,

(b) increasing arrival rates of vehicles destined for a specific point in the road network in order to represent an event at a specified location in an effort to observe how well the algorithms respond to such a traffic flow anomaly,

(c) increasing the length of a road corridor in order to investigate how the scaling of a corridor influences the comparative performance of the algorithms, as well as how the individual performance of each algorithm worsens with an increase in road network size,

(d) varying the distances between neighbouring intersections in a six-intersection road corridor.

V To *suggest* improvements to algorithms found to underperform during the comparison of Objective IV, in order to rectify any shortcomings discovered.

VI To *implement* the improvements of Objective V to the algorithms in the simulation model of Objectives II–III.

VII To *ascertain* the degree to which the suggested improvements of Objective VI implemented in the algorithms lead to increased algorithmic performance.

VIII To *design* new self-organising traffic signal control algorithms and to compare their relative effectiveness with those of the algorithms in Objective IV.

IX To *perform* a statistical comparison of the effectiveness of the traffic signal control algorithms of Objectives IV and VIII in a real-world case study.

X To *suggest* suitable follow-up future work related to the work contained in this dissertation.

The simulation model of Objectives II–III does not incorporate pedestrians into the road network, nor does it include exclusive pedestrian phases. All vehicles are assumed to obey traffic signals and are of a constant length of five metres. Vehicle arrivals are assumed throughout this dissertation to be governed by Poisson distributions. Vehicle collisions or breakdowns are not incorporated, while vehicles only change lanes in order to transition into the correct lane for an upcoming turn, and do not change lanes if they are behind slower vehicles. The road network topologies embedded in the simulation model of Objectives II–III include road corridors of varying lengths, a 3×4 grid of intersections as well as a real-world transportation network model.

1.4 Dissertation organisation

This dissertation consists of a total of ten chapters. The first chapter serves to inform the reader of the importance of alleviating traffic congestion as well as introduce the basic types of traffic signal control. Descriptions and the benefits of self-organisation and emergence in traffic signal control were provided together with a brief problem description. Finally the scope and objectives of the dissertation are described.

Chapter 2 is a literature review in which seven self-organising algorithms from the literature are described in detail. The first algorithm is a rule-based algorithm by Gershenson and Rosenblueth [22] in which traffic signalling follows a set of six rules. The second algorithm is proposed by Lämmer and Helbing [46], and is inspired by pedestrian flows through narrow bottlenecks, while the third algorithm is proposed by Xie *et al.* [87], in which traffic is aggregated into clusters of queues and platoons, making use of stop line detectors as opposed to radar detection. Similarly, the fourth algorithm proposed by Cesme [12] also involves this mode of detection, and is specifically aimed at improving traffic on arterial roads. The last three algorithms are all proposed by Einhorn [16], including an algorithm inspired by the theory of inventory control, an algorithm inspired by the chemical process of osmosis, and finally a hybrid algorithm that combines the first two. The chapter closes by briefly highlighting various advantages and disadvantages of the algorithms as well as stating and motivating which algorithms are to be included in comparisons later on in the dissertation.

Chapter 3 opens with the definitions of common traffic modelling concepts, and this is followed by descriptions of the four distinct types of simulation modelling paradigms. Advantages and disadvantages of simulation modelling are discussed and the necessary steps in building a simulation model are mentioned. Methods of simulation model verification and validation are described, as well as the different types of traffic simulation models and suitable associated software packages available.

The agent-based traffic simulation model built and used in this dissertation is described in detail in Chapter 4. The modelling framework is described, including the method of road construction, implementation of traffic signals and vehicles, as well as the simulation model output. The methods used to verify and validate the model are mentioned, and this is followed by a description of various elements of the experimental design according to which the various traffic signal control algorithms are compared. These include the simulation warm-up period, general specifications of the road network and the statistical tests performed on the simulation model output.

Chapter 5 opens with detailed descriptions of how each of the algorithms are implemented in the simulation model, followed by the results obtained by these algorithms for a 3×4 grid of intersections. Improvements to the algorithms proposed by Einhorn [16] are suggested and implemented, and finally, the statistical results of these improvements are given in the form of graphs, tables and box plots.

Chapter 6 opens with a summary of the parameter settings employed in each algorithm. Two experiments carried out by Einhorn [16] are then reproduced, including a comparison of five self-organising algorithms and a fixed-time control strategy within the context of a four-intersection corridor as well as a 3×4 grid of intersections under both lighter and heavier traffic conditions.

Two new self-organising algorithms are proposed in Chapter 7. These include an algorithm that clusters vehicles into groups, called *platoons*, and an algorithm based on monitoring the road saturation of competing traffic flows. The relative effectiveness of these algorithms are then compared with those of the other five self-organising algorithms and the fixed-time control strategy, also within the context of a four-intersection corridor and a 3×4 grid of intersections under both lighter and heavier traffic conditions.

The relative effectiveness of all the algorithms are compared in the context of four realistic, but hypothetical, scenarios in Chapter 8 and the results are presented in the form of box plots, *post hoc* tables and graphs. The first two scenarios involve the closing of a road in a grid road network, and a varying vehicle arrival rate. The final two scenarios involve a corridor road network in which the effects of the size of the corridor and the distances between consecutive intersections along the corridor are investigated.

The relative effectiveness of the algorithms are also compared in a real-world transportation network model in Chapter 9. This model makes use of actual recorded arrival rates for each manoeuvre along a corridor through eight consecutive intersections.

Chapter 10 is the final chapter of this dissertation, in which a summary of the work presented in the dissertation is given. This is followed by an appraisal of the contributions made, and finally, a number of suggestions are proposed for future work.

CHAPTER 2

Literature review

Contents

In this chapter, seven self-organising traffic signal control algorithms are described in detail. An algorithm developed by Gershenson and Rosenblueth [22], an algorithm developed by Lämmer and Helbing [46] and an algorithm by Xie *et al.* [87] are considered in §2.1. In §2.2, four more recently proposed algorithms by two PhD students are described, including an algorithm by Cesme [12] and three algorithms proposed by Einhorn [16]. The chapter closes with a summary of the state of the art in self-organising traffic signal control algorithms in §2.3.

2.1 Early use of self-organisation in traffic signal timings

The strategy of self-organisation has only relatively recently been employed in traffic signal control algorithms. In this section, the seminal algorithm of Gershenson and Rosenblueth [22] is described as well as an algorithm proposed by Lämmer and Helbing [46]. This is followed by a description of an algorithm proposed by Xie *et al.* [87].

2.1.1 The algorithm of Gershenson and Rosenblueth

Gershenson [21] has claimed that the problem of traffic signal control is more of an adaption problem rather than an optimisation problem, due to typically unpredictable changes in traffic volume. While optimisation yields the best solution, it is typically computationally expensive to compute an optimal solution in the context of traffic signal control for a system that is continuously changing [21]. In 2012, Gershenson and Rosenblueth [22] therefore proposed a heuristic self-organising method of traffic control which consists of six rules. This method is, in fact, an improvement on their earlier *SOTL-platoon* and *SOTL* [14] methods of traffic signal control. Each traffic signal is considered as an agent obeying six rules, with the higher-numbered rules overriding the lower-numbered rules.

The first rule states that at every time step in a discretisation of the time continuum, a counter φ_I is maintained, keeping track of the number of vehicles approaching or already stationary and waiting at a red signal, within a distance d from an intersection I. Once this counter exceeds a predetermined threshold n, the signal will change and the counter is reset to zero. The purpose of this rule is to ensure that signals change when there is a sufficient traffic volume approaching or stationary at a given red signal. Since the cumulative waiting time of vehicles at an intersection is of importance when considering the performance of a traffic signal control algorithm, this quantity serves as a suitable quantity to use in the control of switching signals. Vehicles approaching a red signal will join the queue of stationary vehicles already waiting until a sizeable platoon of vehicles has formed, at which time the signal will turn green. As platoons travel through intersections, their presence may be detected as they approach, giving rise to the formation of so-called *green waves* of vehicles that flow through the system unimpeded.

The second rule states that the minimum duration of a green signal should be at least μ_I time steps. This rule prevents unnecessarily rapid switching of signals during periods of high traffic densities, which can occur when platoons approach an intersection from opposing directions. A second counter $t_{I_{green}}$ is used to record how long a signal has been green at an intersection. The signal may only turn red once $t_{I_{green}} > \mu_I$, at which point $t_{I_{green}}$ is reset to zero.

The third rules states that if the number of vehicles approaching a green signal within a short distance r from the intersection is at most some pre-specified number m, then the signal must remain green (as shown in Figure 2.1). This rule prevents the "tails" of platoons being separated from the rest of the vehicle groups. If, however, $s > m$, then traffic is assumed to be heavy, causing the signals to be changed in which case the platoon is separated.

The fourth rule states that if no vehicle is approaching a green signal within a distance d from an intersection (as shown in Figure 2.1) and at least one vehicle is approaching or has stopped at a red signal within a distance d from the intersection, the signals must be changed. This rule allows for rapid switching of signals during periods of low traffic density, so that the vehicles do not need to wait at intersections in order to form platoons.

The fifth rule states that if a vehicle stops within a short distance e beyond a green signal (as shown in Figure 2.1), the green signal must be changed to red. The purpose of this rule is to prevent vehicles from stopping inside an intersection and obstructing other vehicles attempting to travel through it.

The sixth rule states that if vehicles have stopped in both directions within a short distance e beyond the intersection, both signals should be set to red until one of the directions contains no vehicle within a distance e beyond the intersection. Green time is then allocated to the direction that contains no vehicle within a distance e beyond the intersection. Like the fifth rule, this rule attempts to prevent vehicles from stopping inside an intersection and obstructing oncoming traffic.

2.1.2 The algorithm of Lämmer and Helbing

Another example of self-organisation in traffic signal control was proposed by Lämmer and Helbing [46] in 2008. Their algorithm was inspired by the observation of a natural occurrence of self-organisation found in oscillations of pedestrian flows through narrow bottlenecks. The

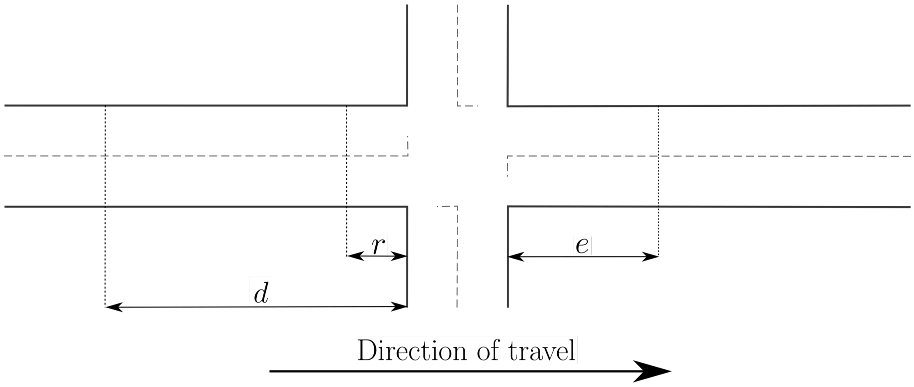


FIGURE 2.1: *A visual representation of the necessary parameters in the algorithm by Lämmer and Helbing [22].*

algorithm makes use of a *stabilisation strategy* and an *optimisation strategy*. While each of these strategies performs poorly under high traffic flow densities, they achieve a far better performance if they are combined appropriately.

Lämmer and Helbing modelled traffic flow using a fluid-dynamic model, considering vehicle flow rates rather than individual vehicle velocities. It is therefore assumed in their model that all vehicles travel at a constant speed. The traffic environment is described in terms of the length L_i, the speed limit V_i and the saturation flow rate Q_i^{max} of road segment i approaching an intersection. The necessary traffic dynamics are defined by an arrival rate $Q_i^{arr}(t) \leq Q^{max}$ as well as a departure rate $Q_i^{dep}(t) \leq Q^{max}$ at time t. These quantities represent the numbers of vehicles per unit time that enter and exit an intersection from road segment i, respectively. Using these flow rates, the accumulated number $N_i^{exp}(t)$ of vehicles expected to travel through the intersection under free flowing traffic conditions at time t is given by

$$N_i^{exp}(t) = \int_{-\infty}^{t} Q_i^{arr}(t' - L_i/V_i)\,\mathrm{d}t',$$

where L_i/V_i is the time taken to travel the distance L_i along road segment i at a speed V_i in free traffic. Due to traffic congestion, the number of vehicles that have actually departed from approach i at time t is

$$N_i^{dep}(t) = \int_{-\infty}^{t} Q_i^{dep}(t')\,\mathrm{d}t' \leq N_i^{exp}(t).$$

Therefore, the difference between $N_i^{exp}(t)$ and $N_i^{dep}(t)$ is the number of delayed vehicles $n_i(t)$, known as the *queue length*. Thus, the queue length is given by $n_i(t) = N_i^{exp}(t) - N_i^{dep}(t)$. While $n_i(t)$ fully encapsulates the associated inflows and outflows of vehicles along approach i, the time required to clear a queue, as well as the associated waiting times, it does not explicitly yield the spatial location of the queue itself.

In this model, setup times are incorporated in order to allow vehicles to clear the intersection safely before an opposing traffic flow is awarded green time. This is achieved by briefly stopping traffic flow in all directions. The setup time τ^0 typically ranges between 3 and 8 seconds. It should also be noted that τ^0 includes the amber phase which accounts for reaction delays as well as delays caused by the initial pull away from a stationary point. The remaining setup time experienced by vehicles is given by $\tau(t)$, where $0 \leq \tau(t) \leq \tau^0$.

The service process can be partitioned into three consecutive phases: the setup time, the time required to clear the vehicle queue and the extended green time that follows after the queue has dissipated. The optimisation strategy implemented by Lämmer and Helbing [46] requires a forecast of the amount of green time $\hat{g}_i(t)$ from the current point in time t that is required to clear the queue in road segment i. The value of $\hat{g}_i(t)$ depends on both the number of queued vehicles $n_i(t)$ and the number of vehicles still joining the queue during the period $\tau_i(t)$. The total time taken for the queue to dissipate is $t + \tau_i(t) + \hat{g}_i(t)$, at which point the total number of vehicles that have arrived at the intersection is equal to the number of vehicles that have departed from the intersection. This gives rise to the conservation of flow law

$$N_i^{dep}(t) + \hat{g}_i(t)Q^{max} = N_i^{exp}(t + \tau(t) + \hat{g}_i). \tag{2.1}$$

In this equation, $\hat{g}_i(t)$ is the largest possible solution. The second term in (2.1) is the number of vehicles travelling through the intersection at the maximum flow rate, which can also be written as $n_i(t)$, since $n_i(t) = \hat{g}_i(t)Q^{max}$. The value of $n_i(t)$ includes all vehicles that are already waiting in the queue, joining the queue during the setup time or clearing it, or arriving in a platoon of vehicles the moment the queue has been cleared.

A "pressure" of priority $\pi_i(t)$ is associated with the approaching flow of traffic along road segment i, such that green time is awarded to traffic flow approaching along road section i if π_i has the highest priority value. The goal is to derive a formula for a priority index π_i in such a way that the signal switching rule acts to minimise the total waiting time of vehicles in both directions of road section i. In order to determine whether it is more beneficial to continue a service phase or to service a different flow direction, it is essential that the total waiting time $\hat{w}_i(t)$ of all vehicles from the stop line to the end of the clearing state, be forecast. The waiting time $\hat{w}_i(t)$ can be expressed as

$$\hat{w}_i(t) = w_i(t) + A_i(t) + B_i(t), \tag{2.2}$$

where, $w_i(t)$ is the total waiting time of all vehicles in road segment i up to time t, $A_i(t)$ is the waiting time experienced by vehicles during the setup time, and $B_i(t)$ is the waiting time incurred by vehicles while the queue is being cleared. The first term in (2.2) is the time integral of the queue length $n_i(t)$ and is expressed as

$$w_i(t) = w_i(t_0) + \int_{t_0}^{t} n_i(t')\,\mathrm{d}t'.$$

The second term $A_i(t)$ in (2.2) may be written as the time integral of the queue length $n_i(t')$ over the time interval $t' = [t, t + \tau(t)]$, that is,

$$A_i(t) = \int_{t}^{t+\tau(t)} \left(N_i^{exp}(t') - N_i^{dep}(t)\right)\,\mathrm{d}t.$$

Since the outflow of vehicles from road segment i during the setup time is zero, $N_i^{dep}(t)$ will be constant with respect to t'. Finally, the third term $B_i(t)$, which represents the clearing time of the queue is a function in which the total number of stationary vehicles along road section i begin departing from the intersection according to a linear function with slope Q^{max}. Therefore,

$$B_i(t) = \int_{t+\tau(t)}^{t+\tau(t)+\hat{g}_i(t)} \left(N_i^{exp}(t') - [N_i^{dep}(t) + (t' - (t + \tau(t)))Q^{max}]\right)\,\mathrm{d}t'.$$

Once again, $N_i^{dep}(t)$ is constant with respect to t'. Since it is necessary to calculate whether it is more effective to keep serving the current traffic flow direction or to change service to a new direction, the rate of increase of waiting time is required, and therefore $\mathrm{d}\hat{w}_i/\mathrm{d}t$ must be determined. It is clear from (2.2) that

$$\frac{\mathrm{d}\hat{w}_i}{\mathrm{d}t} = \frac{\mathrm{d}w_i}{\mathrm{d}t} + \frac{\mathrm{d}A_i}{\mathrm{d}t} + \frac{\mathrm{d}B_i}{\mathrm{d}t}. \tag{2.3}$$

In [45], the steps for calculating the above derivative are omitted and so various additional steps are included in this review in order to verify that the derivation is indeed correct. The first term on the right-hand side of (2.3) is simply

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = n_i(t),$$

as $w(t_0)$ is a constant. The second term on the right-hand side of (2.3), $\mathrm{d}A_i/\mathrm{d}t$, can be separated into two terms, namely

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_t^{t+\tau(t)} N_i^{exp}(t')\,\mathrm{d}t' \tag{2.4}$$

and

$$-\frac{\mathrm{d}}{\mathrm{d}t}\int_t^{t+\tau(t)} N_i^{dep}(t)\,\mathrm{d}t'. \tag{2.5}$$

If the fundamental theorem of the calculus is applied to (2.4), it is found that

$$N_i^{exp}(t+\tau(t))[1+\tau'(t)] - N_i^{exp}(t).$$

The fundamental theorem of the calculus cannot, however, be applied to (2.5) as $N_i^{exp}(t)$ is not a function of t'. It is evaluated instead by first integrating the expression with respect to t' to find

$$-\frac{\mathrm{d}}{\mathrm{d}t}[N_i^{dep}(t)t']\Big|_t^{t+\tau(t)}.$$

After substituting in the integration limits, the result is

$$-\frac{\mathrm{d}}{\mathrm{d}t}[N_i^{dep}(t)\tau(t)],$$

after which the derivative is taken and the final form of this term is

$$-N_i^{dep}(t)\tau'(t) - Q_i^{dep}(t)\tau'(t).$$

Therefore,

$$\frac{\mathrm{d}A_i}{\mathrm{d}t} = N_i^{exp}(t+\tau(t))[1+\tau'(t)] - N_i^{exp}(t) - N_i^{dep}(t)\tau'(t) - Q_i^{dep}(t)\tau'(t). \tag{2.6}$$

Finally, the last term on the right-hand side of (2.3), $\mathrm{d}B_i/\mathrm{d}t$, can be written as

$$\begin{aligned}\frac{\mathrm{d}B_i}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{t+\tau(t)}^{t+\tau(t)+\hat{g}_i(t)} N_i^{exp}(t')\,\mathrm{d}t' - \frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{t+\tau(t)}^{t+\tau(t)+\hat{g}_i(t)} N_i^{dep}(t)\,\mathrm{d}t' - \frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{t+\tau(t)}^{t+\tau(t)+\hat{g}_i(t)} Q^{max}t'\,\mathrm{d}t' \\ + \frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{t+\tau(t)}^{t+\tau(t)+\hat{g}_i(t)} Q^{max}t\,\mathrm{d}t' + \frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{t+\tau(t)}^{t+\tau(t)+\hat{g}_i(t)} \tau(t)Q^{max}\,\mathrm{d}t'.\end{aligned} \tag{2.7}$$

If the fundamental theorem of the calculus is applied to the first term on the right-hand side of (2.7), the expression

$$N_i^{exp}(t+\tau(t)+\hat{g}_i(t))[1+\tau'(t)+\hat{g}_i{}'(t)] - N_i^{exp}(t+\tau(t))[1+\tau'(t)]$$

is obtained. The second term on the right-hand side of (2.7) is simplified by first integrating and then differentiating the term to obtain

$$-N_i^{dep}(t)\hat{g}_i{}'(t) - Q_i^{dep}(t)\hat{g}_i(t).$$

The fundamental theorem of the calculus is once again applied, this time to the third term on the right-hand side of (2.7), which simplifies to

$$-Q^{max}[\hat{g}_i(t) + \tau'(t)\hat{g}_i(t) + t\hat{g}_i{}'(t) + \tau\hat{g}_i{}'(t) + \hat{g}_i{}'(t)\hat{g}_i(t)].$$

The fourth and fifth terms on the right-hand side of (2.7) are simplified by basic integration and differentiation methods to obtain

$$\hat{g}_i(t)Q^{max} + \hat{g}_i{}'(t)Q^{max}t,$$

and

$$\tau'(t)\hat{g}_i(t)Q^{max} + \tau(t)\hat{g}_i{}'(t)Q^{max},$$

respectively. When substituting the last five expressions back into (2.7), various terms cancel, leaving

$$\frac{\mathrm{d}B_i}{\mathrm{d}t} = N_i^{exp}(t+\tau(t)+\hat{g}_i)[1+\tau'(t)+\hat{g}_i{}'(t)] - N_i^{exp}(t+\tau(t))[1+\tau'(t)] - N_i^{dep}\hat{g}_i{}'(t) - Q_i^{dep}(t)\hat{g}_i(t). \tag{2.8}$$

Now that $\mathrm{d}w_i/\mathrm{d}t$, $\mathrm{d}A_i/\mathrm{d}t$ and $\mathrm{d}B_i/\mathrm{d}t$ have been evaluated, the derivative $\mathrm{d}\hat{w}_i/\mathrm{d}t$ in (2.3) can be calculated by summing these three terms to obtain

$$\frac{\mathrm{d}\hat{w}_i}{\mathrm{d}t} = \hat{g}_i(t)Q^{max}[1+\tau'(t)] - Q_i^{dep}(t)[\tau(t)+\hat{g}_i(t)]. \tag{2.9}$$

If the current state is "no service", then $Q_i^{dep}(t) = 0$ and $\tau'(t) = 0$, so that (2.9) becomes

$$\frac{\mathrm{d}\hat{w}_i}{\mathrm{d}t} = \hat{g}_i(t)Q^{max} = \hat{n}_i(t).$$

If, however, the current state is "service", then $\tau(t)$ and $\tau'(t)$ are both zero and $\mathrm{d}\hat{w}_i/\mathrm{d}t = 0$. Therefore,

$$\frac{\mathrm{d}\hat{w}_i}{\mathrm{d}t} = \begin{cases} \hat{n}_i(t), & \text{if road segment } i \text{ is not being serviced,} \\ 0, & \text{during the entire service process of road segment } i, \end{cases} \tag{2.10}$$

as claimed by Lämmer *et al.* [45].

It is necessary to determine when to terminate a specific service period, as doing so will incur an additional setup cost τ^0, causing the predicted waiting time to increase. The magnitude of this increase is denoted by $\Delta\hat{w}_i(t)$ and can be calculated by means of the differential equation (2.10). To determine an expression for $\Delta\hat{w}_i(t)$, the number of vehicles to be served should be a function of both t and $\tau(t)$, in order to observe how the derivative of $w_i(t)$ changes after time $t+\tau(t)$. It is also assumed that the service period is terminated at time t. Since the outflow of vehicles from road segment i satisfies $Q_i^{dep}(t) = 0$, (2.9) leads to the equation

$$\frac{\mathrm{d}\hat{w}_i}{\mathrm{d}(t+\tau(t))} = \hat{n}_i(t,\tau(t)). \tag{2.11}$$

Now $\Delta\hat{w}_i(t)$ can be determined by integrating (2.11) with respect to its second argument from $\tau(t)$ to τ^0 so as to obtain the expression

$$\Delta\hat{w}_i(t) = \int_{\tau(t)}^{\tau^0} \hat{n}_i(t,\tau')\,\mathrm{d}\tau', \tag{2.12}$$

where τ' is the remaining setup time and $\hat{n}_i(t, \tau')$ is the number of vehicles to be serviced at time t. The optimisation approach of the algorithm by Lämmer and Helbing [46] is limited to the expected number of vehicles queued at or approaching the intersection along road section i and it is assumed that there are only two competing directions. From time t, it is expected that for each direction the remaining setup times τ_1 and τ_2, the expected numbers of vehicles $\hat{n}_1$ and $\hat{n}_2$, and the required green times $\hat{g}_1$ and $\hat{g}_2$ be given. It is furthermore assumed that traffic flow 1 is currently receiving green time. Therefore there are two choices:

i the algorithm continues to serve traffic flow 1 before switching to traffic flow 2, or

ii the algorithm switches to traffic flow 2 immediately and will return to traffic flow 1 later, at the cost of an additional setup time.

In order to decide which option minimises the total waiting time, the total increase in waiting time for each option is calculated. If the first option is chosen, then traffic flow 1 will be served for a total of $\tau_1 + \hat{g}_1$ seconds. According to (2.10), the waiting time experienced by traffic flow 2 increases at a rate of $\hat{n}_2$, while the waiting time of traffic flow 1 remains constant. Therefore, the total increase in waiting time as a result of selecting the first option, would be $(\tau_1 + \hat{g}_1)\hat{n}_2$. Alternatively, if the second option were to be chosen, (2.12) yields an additional increase of $\Delta\hat{w}_1$ in the waiting time, which represents the additional setup time required to switch back to traffic flow 1 at a later stage. Traffic flow 2 is served for $\tau_2 + \hat{g}_2$ seconds, while the waiting time of traffic flow 2 increases at a rate of $\hat{n}_1$. Thus, the total waiting time experienced by vehicles is $\Delta\hat{w}_1 + (\tau_2 + \hat{g}_2)\hat{n}_1$. It will therefore be beneficial to continue serving traffic flow 1 as long as

$$(\tau_1 + \hat{g}_1)\hat{n}_2 < \Delta\hat{w}_1 + (\tau_2 + \hat{g}_2)\hat{n}_1. \tag{2.13}$$

This inequality leads to the definition of the priority indices π_1 and π_2 which are obtained by rewriting (2.13) as

$$\pi_1 := \frac{\hat{n}_1}{\tau_1 + \hat{g}_1} > \frac{\hat{n}_2}{\Delta\hat{w}_1/\hat{n}_1 + \tau_2 + \hat{g}_2} =: \pi_2. \tag{2.14}$$

The first priority index π_1 in (2.14) depends solely on variables associated with traffic flow 1, whereas the second priority index π_2 exhibits the same dependence on variables related to traffic flow 2, but with an additional term $\Delta\hat{w}_1/\hat{n}_1$. The penalty for terminating the current service is given by $\Delta\hat{w}_\sigma/\hat{n}_\sigma$, where σ represents the index of the current flow receiving service and it lies in the interval $(0, \tau_\sigma^0)$. Since the penalty is only applied when switching service, the general penalty term $\tau_{i,\sigma}^{pen}$ can be written as

$$\tau_{i,\sigma}^{pen} = \begin{cases} \Delta\hat{w}_\sigma/\hat{n}_\sigma & \text{if } i \neq \sigma \\ 0 & \text{if } i = \sigma. \end{cases} \tag{2.15}$$

Using (2.14) and (2.15), the priority index π_i associated with traffic flow i can therefore be expressed as

$$\pi_i = \frac{\hat{n}_i}{\tau_{i,\sigma}^{pen} + \tau_i + \hat{g}_i}. \tag{2.16}$$

In (2.16), π_i is related to the number of vehicles $\hat{n}_i$ that are expected to proceed through the intersection during a period of $\tau_i + \hat{g}_i$ seconds. The approach that achieves the highest priority index is awarded green time. Local optimisation at each intersection does not necessarily lead to global optimisation of the entire system, as instability in the form of growing queue lengths may occur. In order to combat this instability, a stabilisation strategy is also applied in the algorithm of Lämmer and Helbing [46].

A traffic control strategy is considered stable if vehicle queues remain bounded at all times. Instability in traffic control occurs when signals switch either too frequently, or not frequently enough, resulting in growing vehicle queues. According to Lämmer and Helbing [47], if traffic conditions arise such that vehicle queues remain bounded for a fixed-time control, but not for a local optimisation control strategy, then a fixed-time control is potentially superior to the locally optimised control, and the latter is considered unstable. Figure 2.2 illustrates how queue lengths may grow once a traffic control strategy becomes unstable.

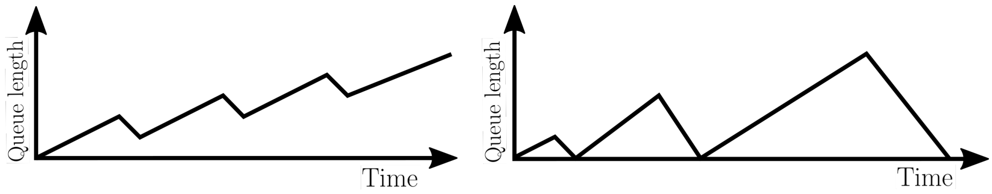


FIGURE 2.2: *An example of two unstable vehicle queues that are growing in length over time.*

Maintaining stability in a system is less like a scheduling problem and more a problem of allocating the appropriate amount of green time to minimise total vehicle delay [46]. In order to maintain stability during local optimisation at intersections, a supervisory mechanism was therefore introduced by Lämmer and Helbing. The purpose of this mechanism is to assess the current traffic conditions and ensure that green times are neither too short nor too long in order to avoid growing queue lengths. The following stabilisation rule was suggested by Lämmer and Helbing [46]: Let Ω be the set of traffic flows chosen by the supervisory mechanism that require imminent service so as to prevent instability. It is assumed that the traffic flow along road section i joins the set Ω once the capacity of road section i has reached a critical number of vehicles, denoted by n_i^{crit}. The parameter n_i^{crit} is specified such that the following two safety requirements are met:

1. each traffic flow must be served once, on average, within a desired interval of length Z, and
2. each traffic flow must be served at least once within a maximum service interval of length Z_{max}, such that $Z_{max} \geq Z$.

In the above requirements, Z is the cycle time of an associated stable, fixed-time control programme and Z_{max} is the maximum red time that a traffic flow can experience. The anticipated service interval $z_i(t)$ of traffic flow i consists of the setup time, the preceding red time and the anticipated green time, *i.e.* $z_i(t) = \tau_i^0 + r_i + \hat{g}_i$. Assuming that the average arrival rate of vehicles, cycle lengths and green times of a stable, fixed-time control programme are given, a threshold function for n_i^{crit} can be derived as

$$n_i^{crit}(z_i(t)) = Q_i^{arr}(t)Z\frac{Z^{max} - z_i(t)}{Z^{max} - Z}. \tag{2.17}$$

This function meets the first requirement above, since when $z_i(t) = Z$ (the anticipated service interval has the same length as the desired service interval), the number of vehicles expected to arrive is $Q_i^{arr}(t)Z$. The second requirement is also met, because when $z_i(t) \geq Z^{max}$ (the anticipated service interval is equal to or larger than the maximum red time), $n_i^{crit}(z_i(t)) \leq 0$. Once an approach has been classified as critical ($n_i(t) \geq n_i^{crit}(z_i(t))$), the approach is added to the set Ω. The first element in Ω is the first to receive service and will continue to receive service until the queue has dissipated, or until the traffic flow receives a green time duration that would be awarded by a fixed-time control programme, which is equivalent to $\frac{Q_i^{arr}(t)}{Q_i^{max}}Z$. While Ω is nonempty, the stabilisation strategy is implemented and the necessary flow will receive green time. When Ω is empty, however, the optimisation strategy described above is followed according to the priority indices π_i.

In summary, the index of the road section receiving service according to the two strategies described above is given by

$$\sigma = \begin{cases} \text{head of } \Omega, & \text{if } \Omega \neq \emptyset, \\ \max_i \pi_i, & \text{otherwise.} \end{cases} \tag{2.18}$$

When $\Omega = \emptyset$, the optimisation strategy is implemented, which aims to minimise vehicle delay by serving the approaching vehicles as fast as possible. On the other hand, when $\Omega \neq \emptyset$, the stabilisation strategy takes over in order to avoid the situation where queues grow too long and exceed the threshold n_i^{crit}. Therefore, the optimisation strategy has full control of the system, unless it becomes unstable, in which case the stabilisation strategy is implemented in order to regain stability and allow the optimisation strategy once again to commence.

2.1.3 The algorithm of Xie *et al.*

In 2011, Xie *et al.* [87] proposed a *platoon-based self scheduling (PBSS)* algorithm in which intersections are considered independent of one another and function with a restricted visibility of incoming traffic. The algorithm aggregates approaching traffic into clusters of anticipated queues and platoons based on similarities in traffic flows.

Vehicle speed and average vehicle delay time are typically used as algorithmic performance indicators to measure the flow of traffic under the control of a specific algorithm. In the algorithm of Xie *et al.* [87], it is assumed that all road sections have fixed length L, where $L \in [250, 500]$, and that all intersections consist of two one-way roads. Each road is assumed to have a stop-line detector as well as an upstream detector located a distance L_{det} upstream from the intersection, where $L_{det} < L$.

Vehicle kinematic data are collected over a detection window by periodically sampling from the upstream detector in order to obtain information about approaching vehicles. This sampling returns the quintuple $(\tau_{ps}, \tau_{pd}, \tau_{pe}, n_{pc}, q_{pc})$ containing the start time τ_{ps} of the sample period, the duration τ_{pd} of the period, the end time τ_{pe} of the sample period, the number of vehicles n_{pc} that have been counted over the sample period and the vehicle flow rate q_{pc}. Instead of dealing with the actual start time of a sample period, the tuple offset

$$\tau_{ps}(t) = \tau_{ps} - t - L_{det}/v_f \tag{2.19}$$

of the intersection is used. In (2.19), v_f is the expected free flow speed of approaching vehicles and $\tau_{ps}(t)$ is the expected time taken by the vehicle that was counted first over the sampling period to reach the intersection. Tuples containing positive values of n_{pc} are added to an ordered sequence, and this sequence is considered at various decision points in time. Vehicles that depart from the intersection are monitored by the stop-line detector in order to keep track of the current queue length at the intersection for approaching road sections, denoted by n_q. Each time a vehicle crosses the stop-line detector, the queue count is decremented by 1.

At the next decision point, tuples collected over the detection window are aggregated into clusters, by merging tuples that have duration values within a specific threshold c_{thc} into an ordered sequence S. The new tuple formed takes the minimum value of the clustered tuples as start time τ_{ps}, the sum of the tuple durations as the new period duration τ_{pd} and the sum of the vehicle counts as the new vehicle count n_{pc}. The new end time τ_{pe} is recalculated as $\tau_{pe} = \tau_{ps} + \tau_{pd}$ and the new flow rate is recalculated as $q_{pc} = n_{pc}/\tau_{pd}$. The merged clusters are then classified, according to their sizes and positions, into one of three categories. The first category is a *queue cluster*, which consists of vehicles that are expected to have already arrived at the intersection. The second category is a *platoon cluster*, which consists of vehicles that are not expected to have arrived at an intersection, have a vehicle count of more than a certain threshold ($n_{pc} > c_{thpc}$) and have a flow rate greater than a certain threshold ($q_{pc} > c_{thpd}$). The remaining clusters make up the third and final category, called *minor clusters*.

The above-mentioned clusters are used to estimate the queue length at each of the intersection approaches. In order to estimate these queue lengths accurately, it is important to compute how long it will take for a queue to clear. The queue clearing time is calculated as

$$\tau_{qc}(n_q, \tau_{ge}) = \begin{cases} \tau_{sh} \cdot n_q + \tau_{sl} - \tau_{ge} & \text{if } \tau_{ge} < \tau_{sl} \\ \tau_{sh} \cdot n_q & \text{if } \tau_{ge} \geq \tau_{sl}, \end{cases} \tag{2.20}$$

where n_q is the number of queued vehicles, τ_{ge} is the time that the signal has been green during the current phase, τ_{sl} is the time lost during vehicle pull away and τ_{sh} is the saturation headway between vehicles. Based on the value of τ_{qc}, the anticipated number of vehicles in the queue n_q together with the vehicles that are expected to arrive before the current queue is cleared may be calculated by means of Algorithm 2.1 This algorithm requires as input the number of vehicles n_q currently stopped at the intersection, the elapsed green time τ_{ge} and the future time τ_{adv} for which the queue should be calculated. The output of this algorithm is the number of vehicles at the intersection at a particular time or are expected to arrive at the intersection before the queue has dissipated.

Algorithm 2.1: Estimate n_{qa} in the cluster sequence S.

Input : n_q, $\tau_{ge} = 0$ or $\tau_{adv} = 0$.
Output: Anticipated queue length n_{qa}

$n_{qa} \leftarrow n_q$
for $i = 0$ *to* $Size(S)$ **do**
  Obtain $\tau_{qc}(n_{qa}, \tau_{ge})$ using (2.20)
  if $(\tau_{ps,i} - \tau_{adv}) \leq \tau_{qc}$ **then**
    $\Delta_d \leftarrow 1/\tau_{sh} - q_{pc,i}$
    if $\Delta_d \leq 0$ *or* $(\tau_{pe,i} - \tau_{adv}) \leq \tau_{qc}$ **then**
      $n_{qa} \leftarrow n_{qa} + n_{pc,i}$
    else
      $\Delta_t \leftarrow (\tau_{qc} - (\tau_{ps,i} - \tau_{adv})) \cdot q_{pc,i}/\Delta_d$
      if $\Delta_t < \tau_{pd,i}$ **then**
        $n_{qa} \leftarrow n_{qa} + n_{pc,i} \cdot \Delta_t/\tau_{pd,i}$
        return
      else
        $n_{qa} \leftarrow n_{qa} + n_{pc,i}$

The next step in the PBSS algorithm is to decide whether to continue serving a certain phase or to terminate it and start serving the next phase of the cycle. Three different policies are defined that have the potential to extend the green signal currently being displayed. The policies are ordered according to priority precedence and if a policy returns a value that is not zero, this value becomes the extended amount of green time. If a policy returns a value of zero, on the other hand, the next policy is considered. If all policies return a value of zero, the current green signal is terminated. The three policies, in order of priority, are *anticipated all clearing (AAC)*, *platoon-based extension (PBE)* and *platoon-based squeezing (PBS)*.

The AAC policy is based on an approach proposed by Lämmer and Helbing [46] which involves investigating whether any queues are anticipated at the intersection currently receiving service. If there is indeed an anticipated queue, the AAC policy attempts to accommodate the situation by extending the green signal until the queue has dissipated, given that a pre-specified maximum green time has not yet been reached. The logic of this policy is described in Algorithm 2.2.

Algorithm 2.2: Anticipated all clearing (AAC) policy.

Input : The phase index k, and the elapsed green time τ_{ge}.
Output: Green time extension τ_{ext}.

```
1 Estimate n_qa by means of Algorithm 2.1, given τ_ge and τ_adv = 0
2 Obtain τ_ext ← τ_qc(n_qa, τ_ge) using (2.20)
3 τ_ext ← min(τ_ext, remaining variable time at k)
4 return τ_ext
```

The PBE policy aims to prevent the separation of platoons travelling through an intersection. This is achieved by detecting platoons as they approach the intersection and attempting to extend the current green phase in order to allow the entire platoon to pass through the intersection as a whole. Clusters can be partitioned into three different groups: *queue clusters, platoon clusters* and *minor clusters*. Queue clusters are expected to have reached the intersection, while platoon clusters are not expected to have reached an intersection and have a vehicle count above a certain threshold. The remaining clusters are referred to as minor clusters. Along the approach currently receiving green time, the number of vehicles in the platoon is denoted by $n_{p,g}$, the start time of the green signal is denoted by $\tau_{ps,g}$, the end time of the green signal is denoted by $\tau_{pe,g}$ and the total number of vehicles in the minor clusters ahead of the platoon is denoted by $n_{m,g}$. Along the approach currently receiving a red signal, the number of queued vehicles that are anticipated to arrive before the queue is cleared is denoted by $n_{qa,r}$ (calculated using Algorithm 2.1), while the anticipated queue size predicted for $\tau_{adv} = \tau_y$ is denoted by $n'_{qa,r}$ and the number of vehicles in the minor clusters is denoted by $n_{m,r}$. Similarly, $n'_{m,r}$ represents the anticipated number of vehicles in the minor clusters at $\tau_{adv} = \tau_y$, and $\tau'_{qa,r}$ denotes the anticipated clearing time of the anticipated queue at time τ_y to clear $n'_{qa,r}$. In order to determine whether a green signal should be extended, the logic in Algorithm 2.3 is followed, as illustrated in Figure 2.3.

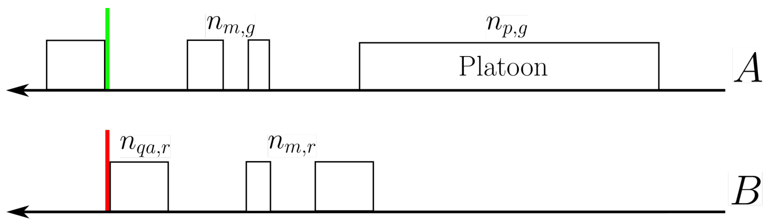


FIGURE 2.3: *An example of platoon-based extension. The red and green line represent the current signal displayed for the flow at the intersection.*

In the first three lines of Algorithm 2.3, the green time required to facilitate the dissipation of the expected queue along an approach currently receiving a red signal is calculated by means of Algorithm 2.1 and (2.20). In line 4 of the algorithm, the maximum idle green time for the direction receiving a green signal is estimated. The difference between the total switching time

Algorithm 2.3: Platoon-based extension (PBE) policy

Input : An estimate of $n'_{qa,r}$ using Algorithm 2.1, and $\tau_{adv} = \tau_y$.
Output: Green time extension τ_{ext}.

1 $\tau'_{qa,r} \leftarrow \tau_{qc}(n'_{qa,r}, 0)$, using (2.20) for τ_{qc}
2 $\tau'_{qa,r} \leftarrow max(\tau'_{qa,r}, \text{remaining varible time})$
3 $\tau_{idle,g} \leftarrow \tau_{ps,g} - \tau_{qc}(n_{m,g}, 0)$ using (2.20) for τ_{qc}
4 $\Delta_\tau \leftarrow (\tau'_{qa,r} + 2 \cdot \tau_y) - \tau_{idle,g}$
5 **if** $\Delta_\tau > 0$ **then**
6 $\quad n'_{m,r} \leftarrow n_{m,r} + n_{qa,r} - n'_{qa,r}$
7 $\quad \delta_r \leftarrow n'_{m,r} \cdot \Delta_\tau - n'_{qa,r} \cdot (\tau_{pe,g} + n'_{m,r} \cdot \tau_{sl})$
8 $\quad \delta_g \leftarrow (\Delta_\tau + \tau_{sl}) \cdot (n_{p,g} + n_{m,g}) + n_{m,g} \cdot \tau_{idle,g}/2$
9 $\quad$ **if** $\delta_g + \delta_r > 0$ **then**
10 $\quad\quad$ **return** $\tau_{ext} \leftarrow \tau_{pe,g}$
11 **return** $\tau_{ext} \leftarrow 0$

and the idle green time is calculated in line 5. If the idle green time is shorter, then the total delay time for both directions is estimated in lines 7–10 and the output of the algorithm is based on that result. PBE differs from AAC in the way that it is willing to allow idle green time in order to allow an approaching platoon to pass through the intersection instead of forcing it to wait for the next cycle, as this has the potential to result in a smaller total vehicle delay.

The PBS policy also aims to best serve platoons of vehicles, but specifically those platoons along approaches that are receiving a red signal. This policy first tests whether a platoon is approaching along a direction that is currently receiving a red signal. If such a platoon is detected, the PBS determines whether it can extend the current phase in order to ensure a transition to a green signal that best suits the approaching platoon (avoids separation of the platoon). The PBS policy (illustrated in Figure 2.4) decides whether or not the current green phase should be extended to the maximum allowed idle time, and this logic is conveyed in Algorithm 2.4.

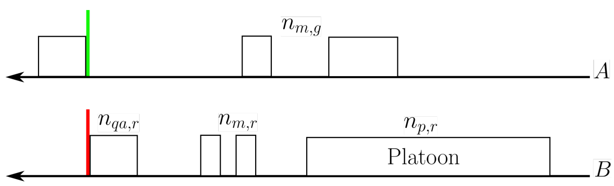


FIGURE 2.4: *An example of platoon-based squeezing, where the number of vehicles in a platoon receiving a red signal is represented by $n_{p,r}$.*

The performance of PBSS was tested by Xie *et al.* [87] within a simulated environment built using the open source microscopic traffic simulator, *Simulation of Urban Mobility* (SUMO). Two different road network topologies were considered, namely an arterial road consisting of five intersections and a 2×3 grid of six intersections. The PBSS was compared against two other self-organising algorithms as well as a fixed control scheme. These algorithms were compared in respect of the total delay times they induce in the network and the average speed of vehicles. The upstream detectors L_{det} were assumed to be located $L - 50$ metres from the intersection and the amber and all-red phase τ_y was assumed to be 5 seconds, while the minimum green time τ_g^{min} and the maximum green time τ_g^{max} were taken as 5 seconds and 55 seconds, respectively.

Algorithm 2.4: Platoon-based squeezing (PBS) policy

Input : $\tau_{qc}(n_{qa,r} + n_{m,r}, 0)$ using (2.20)
Output: Green time extension τ_{ext}.

$\tau'_{qa,r} \leftarrow \tau_{qc}(n_{qa,r} + n_{m,r}, 0)$
$\tau'_{qa,r} \leftarrow max(\tau'_{qa,r},$ remaining variable time)
$\tau_{idle,r} \leftarrow \tau_{ps,r} - \tau'_{qa,r} - \tau_y$
if $\tau_{idle,r} < (\tau_{sg}^{min} + 2 \cdot \tau_y)$ **then**
    return $\tau_{ext} \leftarrow \tau_{idle,r}$
return $\tau_{ext} \leftarrow 0$

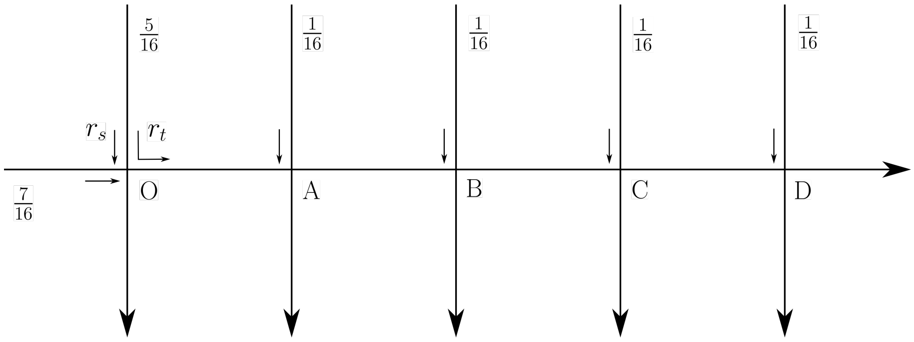


FIGURE 2.5: *Road network topology implemented in SUMO by Xie et al. [87]. The network consists of an artery with five intersections.*

The first road network topology considered was the arterial road depicted in Figure 2.5. Traffic flow rate proportions were assumed to be 1/16 for each of the cross-street approaching intersections A, B, C and D, while that of cross-street at O was taken as 5/16 and the through phase approaching intersection O was assumed to be 7/16. Vehicle turns were modelled to occur at intersection O with probability $r_t/(r_s + r_t)$, where r_t is initially zero. The simulation was run over a period of one hour, increasing r_t every 20 minutes by Δr_t while $r_s + r_t = 5/16$.

The second road network considered by Xie *et al.* [87] was the grid of intersections shown in Figure 2.6. There are considerably more route options in this network than in the previous network, but only eight routes were considered. The five routes straight through the network each contributes 1/12 of incoming traffic, collectively generating 5/12 of the total traffic into the road network. The other three routes shown in Figure 2.6 account for the remaining incoming traffic inflow of 7/12.

Three different versions of PBSS were implemented by Xie *et al.* [87] and compared to three other existing algorithms. The primary version, PBSS, makes use of both platoon-based extension and platoon-based squeezing, while another version of PBSS uses only PBE and the final version of PBSS uses only PBS. The flow velocity v_f, the time lost during vehicle pull away τ_{sl} and saturation headway τ_{sh} were assigned realistic values according to historical data. The following values were assigned: $v_f = 0.95v_{max}$, where v_{max} is the speed limit of 10 metres per second, $\tau_{sl} = 3$ seconds and $\tau_{sh} = 3$ seconds. Xie *et al.* also took the threshold duration as $c_{thc} = 5$ seconds, the vehicle number threshold as $c_{thpc} = 5$ vehicles and the threshold flow rate as $c_{thpd} = 1/c_{thc} = 1/5$ vehicles per second.

As mentioned, PBSS was compared to three other algorithms. The first algorithm is known as *Self-Organising Traffic Lights (SOTL-phase)* [21], which is an algorithm comprising the first

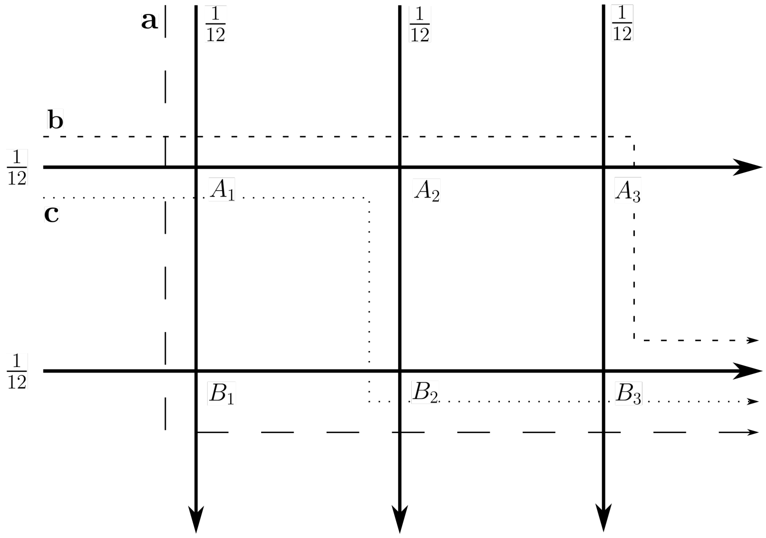


FIGURE 2.6: *Road network topology implemented in SUMO by Xie et al., consisting of a 2×3 grid of intersections. The five minor flow straight routes are displayed as well as the three routes* **a**, **b**, *and* **c** *with major traffic flows.*

two steps of the algorithm by Gershenson and Rosenblueth, described in §2.1.1. The counter threshold n in the first step in §2.1.1 was set to the default value of 41 and the minimum duration μ_I in the second step in §2.1.1 was set to the default value of 20.

The second algorithm against which PBSS was compared is the so-called AAC policy, which is based on the method implemented by Lämmer and Helbing in [46], described in detail in §2.1.2.

The third algorithm is a fixed-time control scheme that aims to maximise the average vehicle speed when $r_t = 0$. Each intersection has a fixed cycle of 70 seconds for both road network topologies. Intersection O in the arterial network was assigned green time splits of 25 seconds for cross traffic and 35 seconds for artery traffic, which is in line with the proportions of incoming traffic from those directions. Intersections A–D in the artery were assigned different splits depending on the length of L. If $L = 250$, cross-street traffic received 17 second splits, while artery traffic received 43 second splits and the offset between signals at neighbouring intersections was taken as 28 seconds. When $L = 500$, on the other hand, cross-street traffic received 19 seconds of green time, while artery traffic received 41 seconds of green time and the offset between intersections was taken as 54 seconds.

Three scenarios were implemented and tested in the simulation. In the first scenario, it was required that all intersections in the arterial road network are adaptively controlled. The second scenario is the same as the first, except that intersection O is under a fixed control scheme. This scenario tests each controller's ability to recover from bottleneck traffic that will be experienced at intersection O. The final scenario involves the grid network, with each intersection being controlled adaptively. In each of these scenarios, $\Delta r_t \in \{0, 1/16, 2/16\}$.

In the first scenario where the arterial road network is subjected to adaptive control at all intersections, the PBSS was reported to outperform all three other algorithms with respect to both highest average vehicle speed and lowest average waiting delay, for both $L = 250$ and $L = 500$, for all three values of Δr_t.

In the second scenario, the algorithms were compared with respect to the arterial waiting time ($\tau_{w,art}$) which emanates along the arterial road's approaching intersections A–D and the non-bottleneck waiting time ($\tau_{w,nb}$) which emanates along the cross-street roads leading to intersections A–D. In this scenario it was found that as Δr_t increases, the performance of PBSS relative to the other algorithms, increases the most. When $L = 250$ and $L = 500$, the arterial waiting time for the fixed control scheme was the lowest when Δr_t was set to zero, since no turns occurred at intersection O. Since there was no change in the traffic flow, the fixed algorithm performed the best, although the PBSS still achieved the lowest cross-street waiting time for all values of Δr_t. As Δr_t increased, however, so did $\tau_{w,art}$ for fixed time control. While the performance of the fixed algorithm worsened as it struggled to adapt to the changes in traffic flow, $\tau_{w,art}$ did not worsen significantly for PBSS and performed better than the fixed-time control in terms of $\tau_{w,art}$ once Δr_t had increased. Overall, PBSS was once again found to perform the best in this scenario.

In the final scenario, the algorithms PBSS, PBSSe and PBSSs were compared to SOTL-phase and AAC in the context of the grid road network, with $L = 500$ for low, medium and high traffic demands. The three PBSS methods significantly outperformed AAC and SOTL-phase methods for all traffic densities. In comparison with one another, PBSSs achieved a higher average speed than PBSSe, while PBSS performed the best overall once again.

While the PBSS algorithm appears promising in terms of reducing vehicle delay time and increasing vehicle speed, these results were obtained under the assumption that all roads are one-way and that each intersection lies at a fixed distance from its neighbouring intersection. Some threshold parameters in this algorithm were assigned specific values without clear motivation and it is also noted that in the two topologies considered, barely any turning was taken into account, which is an unrealistic assumption.

2.2 More recently proposed self-organising algorithms

This section opens in §2.2.1 with a description of a self-organising algorithm proposed by Cesme [12] which prioritises traffic surrounding arterial roads. This is followed by descriptions of three self-organising algorithms proposed by Einhorn [16]. In particular, an algorithm inspired by the theory of inventory control is reviewed in §2.2.2, while an algorithm inspired by the chemical process of osmosis is considered in §2.2.3. A description of a hybrid algorithm, which is a combination of the last two algorithms, finally follows in §2.2.4. Each of the three self-organising algorithms of Einhorn [16] requires the presence of radar detection equipment mounted at intersections.

2.2.1 An algorithm proposed by Cesme

Cesme [12] focused on the traffic signal control problem associated with arterial roads in the belief that the use of self-organisation is the most effective way to alleviate traffic due to the adaptive and responsive nature that is inherent in self-organising agents. Both a stop-line detector and an upstream detector are again assumed to be the mode of detection of vehicles approaching the intersection. The stop-line detector is used to register a vehicle call and initiate a green signal, while the upstream detector extends the green time until a vehicle *gap-out* occurs. Such a vehicle gap-out occurs when there is an absence of detected vehicles over a specified period of time. The length of this specified gap is called the *critical gap*.

Cesme [12] claimed that it is important to select a suitable critical gap as it needs to be long enough to allow a queue to clear, but short enough to avoid wasting green time. The notion of a non-simultaneous gap-out is used in the signal switching logic. This refers to signals changing once a gap-out occurs in both directions currently receiving a green signal, as opposed to only

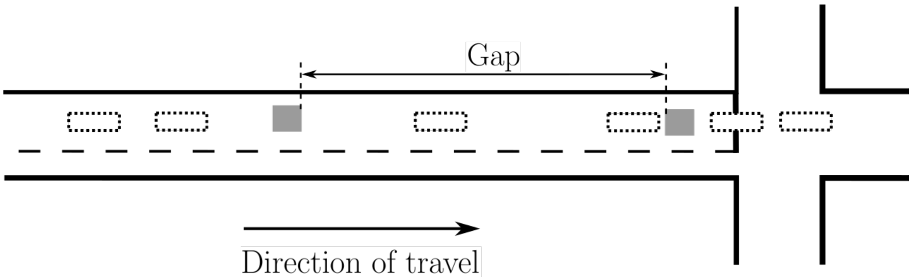


FIGURE 2.7: *Stop-line and upstream vehicle detection. Vehicles travel through the intersection while the upstream detector extends the green time until there are no longer any vehicles between the two detectors, at which point the stop-line detector initiates a signal change.*

switching signals when both directions reach a gap-out simultaneously. This prevents unnecessarily long and wasted green times. The equation

$$h = Gap + \frac{L_d + L_v}{v}$$

relates the gap time *Gap* to the time headway h, L_d is the length of the detector, L_v is the length of the vehicle and v is the desired speed of the vehicle. Therefore, the *critical time headway* is merely the critical gap time together with the time taken for a vehicle to travel the length of the detector. A *critical intersection* is an intersection with the longest natural cycle time, which varies as traffic conditions change over time.

A number of different signal control strategies have been proposed to promote coordination between intersections and encourage self-organising behaviour, specifically for arterial roads. These strategies include rules for *secondary extension* at widely spaced intersections and dynamic coordination logic at closely spaced intersections.

A secondary extension is an extended period of green time awarded to a phase once the initial allocated green time has elapsed and a gap-out has occurred. This is usually necessary when a platoon of vehicles is about to arrive at an intersection, just after the approach phase gaps out. In order to decide whether to grant a secondary extension, it is necessary to exchange information on approaching platoons between two neighbouring intersections, an upstream intersection i, and a downstream intersection j. The following two steps are followed:

1. As the signal turns amber before the start of a green time at intersection i, the number $n_{Q,i}$ of queued vehicles between the upstream detector and the stop-line detector is recorded and communicated to intersection j. Let t be the time at which the amber phase begins at intersection i and let $A_k(t)$ be the expected arrival time of vehicle k at intersection j, where
$$A_k(t) = t + Y_i + T_{ij} + k \cdot H_{sat_i}$$
for all $k = 1, \ldots, n_{Q,i}(1 - P_{RT,i})$. Here $P_{RT,i}$ is the proportion of vehicles turning right at intersection i, Y_i is the length of the amber phase, T_{ij} is time taken to travel from intersection i to intersection j and H_{sat_i} is the saturation headway of vehicles travelling straight through intersection i.

2. While intersection i is receiving green time, the detector upstream from it updates the arrival profile anticipated at intersection j.

In order to decide whether or not to provide a secondary extension after a phase gap-out, a number of attributes of the approaching platoon have to be determined. These attributes include the number of vehicles in the platoon (a larger number of vehicles will benefit if there is an extension), the density of the platoon (denser platoons result in less green time being wasted if there is an extension) and imminence (the closer the platoon is to the intersection, the less delay time other queued vehicles experience). These three parameters are incorporated into a variable L^*, which is "minimum ratio of lost time during a tentative secondary extension to the number of arrivals during that tentative secondary extension, minimised over different potential lengths of secondary extension" [12]. Assume that a phase has just gapped-out ($t = 0$) and let $L(t)$ be the time lost per vehicle if the secondary extension time is t. Therefore,

$$L(t) = \frac{t}{n(t)} - h_{sat} \tag{2.21}$$

and

$$L^* = \min_{t=2,4,\ldots,S_{max}} L(t),$$

where $n(t)$ is the number of vehicles expected to travel through the intersection if a secondary extension of length t were to be granted, which is known from the anticipated arrival profile of the preceding intersection. The first term on the right-hand side of (2.21) is the average time headway and may be denoted by h_{avg}, while the saturation time headway is denoted by h_{sat} and S_{max} is the maximum allowed length of a secondary extension. $L(t)$ is calculated for various values of t and the value of t which minimises $L(t)$ is the time selected for the secondary extension. For very low values of t, $L(t)$ is typically large as few vehicles are expected directly after the gap-out, while in the case where a gap-out is followed by a platoon of vehicles, L^* will be minimised at the time when the last vehicle in the platoon reaches the stop-line. The larger, denser and more imminent the platoon is, the smaller L^* will be.

Cesme [12] stated that excess capacity during under-saturation of an intersection may be treated as a resource that can be allocated as the controller wishes, taking into account that the more saturated an intersection is, the less likely it should be to grant a secondary extension in order to avoid wasted green time. In order to decide whether a secondary extension should be granted to an approaching platoon, excess capacity of an intersection is taken into account. A secondary extension is only granted if the lost time per vehicle L^* is less than or equal to the "affordable lost time." This value is a function of an intersection's volume to capacity ratio, which is given by

$$v/c = \frac{\sum_{critical\ i} \frac{v_i}{s_i}}{1 - L/C},$$

where v_i and s_i represent the arrival rates and saturation flow rates, respectively, summed over critical movements only, while C is the maximum allowed cycle length and L is the total time lost for the critical movements. As the ratio v/c grows, the "affordable lost time" tends to zero and it becomes increasingly difficult for a secondary extension to be granted. Once an intersection becomes saturated, no secondary extension is granted. Cesme [12] decided on a cut-off time of two seconds of lost time per vehicle for small values of v/c. The affordable lost time per vehicle is given by

$$A_{LT} = \min\left\{L_{\max}, \left(\frac{\text{critical } v/c \text{ ratio}}{\text{intersection } v/c \text{ ratio}} - 1\right) \cdot L_{\max}\right\},$$

where the critical v/c ratio is specified by Cesme [12] as 1.0, indicating that the intersection will function at its capacity if the critical ratio is reached. It is important to note that a secondary extension is only granted if it does not result in over capacity.

In order to determine whether a phase should receive a secondary extension and, if so, what the length of this potential extension should be, the lost time per vehicle is calculated as well as the affordable lost time per vehicle. If the lost time per vehicle is less than the affordable lost time, then a secondary extension is granted with a duration of t^*, the time which minimises L^*.

Provided that a secondary green extension is granted, an important question is how much the maximum allowed additional green time S_{max} should be for a certain phase in a cycle. Cesme suggests a value of $S_{max} = 20$ seconds for a critical through-movement, a value which was obtained through a sensitivity analysis [12]. For a non-critical movement, on the other hand, the stricter maximum

$$S_{\max} = \min\{\max 10, \Delta C_n, 20\}$$

is specified for a secondary extension, where ΔC_n is the difference between the natural cycle lengths of two neighbouring intersections. If a particular intersection has a longer natural cycle than any of its neighbours, then $\Delta C_n = 0$ for that intersection. A secondary extension for non-critical movements is restricted to 10 seconds when the cycle length is close to those of neighbouring intersections, because then an extension is only granted to platoons with small lost times per vehicle. If, however, a neighbouring intersection has a much longer natural cycle length than the local intersection, a secondary extension time is granted, as it increases progression for both the non-critical and critical direction by developing more similar cycle length times between adjacent intersections.

When it comes to intersections that lie within a short distance from one another, additional precautions have to be taken to ensure that queue spillback to adjacent upstream intersections does not occur. Closely spaced intersections are taken to be adjacent intersections that lie within approximately 180 metres from one another, requiring special control logic in order to operate effectively.

A coupled zone consists of a small number of intersections. Coupled logic is therefore not suitable for city grids containing a large number of closely-spaced intersections. Each of these zones is controlled by the critical intersection in that zone, which is calculated during each cycle. Coordination within these zones aims to facilitate green waves through the critical movement by giving priority to the critical direction in the zone. When there is little risk of oversaturation (*i.e.* when $v/c < 0.9$) it is beneficial for intersections in the same coupled zone to have simultaneous green starting times as it promotes good coordination in both directions due to the small spacing between intersections. When there is a substantial risk of oversaturation (*i.e.* $v/c > 0.9$), the controller prioritises the critical movement, using offsets to ensure that the downstream intersections are cleared in time for the approaching platoon to travel through unobstructed.

The "earliest activation time" for each phase at a critical intersection is estimated at every phase transition. This time is based on the current traffic signals, minimum green signal times and the minimum time required to clear current queues based on the n_Q-value of the critical intersection. The earliest activation times are exchanged with neighbouring intersections in a coupled zone which exchange it with their neighbours until the information has reached all the intersections in the zone. When $v/c < 0.9$ in a critical intersection, the earliest activation time then becomes the actual "scheduled activation time" of the arterial through phase for non-critical intersections in the coupled zone.

Usually, non-critical intersections are prepared to begin serving the arterial through phase before the earliest activation time, because the natural cycle length of a critical intersection is longer than its neighbours. When a cross-street (a street perpendicular to an arterial road) phase preceding the arterial through phase gaps out, the green signal is held until the earliest activation time of the arterial through phase is reached. The time from the gap-out to the activation time is known as the *slack* and it is seen as additional green time that may be allocated to either extend the previous phase or start the next phase sooner. By holding the green signal until the earliest activation time is reached, this mechanism reduces delay of cross-street traffic, as certain vehicles will not be forced to wait for the next cycle before they cross the intersection. This also aids in preventing non-critical intersections from becoming critical.

Left turn phases may be classified as either leading or lagging left phases. A *leading* left phase is a left turn phase that precedes the opposing through phase, while a *lagging* left phase refers to a left turn which follows the opposing through phase. In a coupled intersection with a leading left phase preceding an arterial through phase, the slack time required to hold this left until the activation time of the through phase is an inefficient use of slack time unless there is an unexpectedly large number of vehicles turning left. In such a case, it would be more beneficial to ignore the previous gap-out and continue serving the previous cross-street phase with the slack time. The necessary split time required for the leading left phase is calculated using the number of vehicles in the queue together with an arbitrarily chosen 10% so as to account for additional arriving vehicles as well as for vehicles that depart from the intersection slower than expected.

As a platoon departs from an intersection upstream in a coupled zone, the controller ensures that at the following downstream intersections green signals are held until the platoon has travelled through them and has left the coupled zone. Some further rules for coupled intersections are added in order to ensure good coordination. One of these rules states that a critical intersection situated upstream from any non-critical intersections in the same coupled zone has the authority to end any of the cross-street green signals downstream from it. This prevents the scenario occurring in which a downstream non-critical intersection extends the predicted cross-street phase due to slower departing vehicles or an extensive queue length that reaches past the upstream detector. In this scenario, if a large platoon is released early from an upstream intersection along the arterial through phase, reaches the downstream intersection and is stopped by a red signal, then there is a chance that spill-back may occur and that the critical intersection may become oversaturated.

Another technique used to improve two-way coordination between coupled zones involves “lead-lag configuration” which refers to having a leading left turn at one intersection and a lagging left turn at the next. This limits green time wasted while waiting for vehicles arriving at the next intersection as this time may be used to serve left-turning vehicles at the next intersection. It has been shown that for closely spaced intersections, a lead-lag strategy is more efficient than a strategy involving leading left phases only [13].

Secondary extension logic in a critical direction within a coupled zone is applied only at the first intersection of the zone due to the communication between intersections that allows for information to be passed downstream. If a secondary extension is granted, this information is received by downstream intersections so as to ensure that green signals are not terminated before the approaching platoon from the upstream intersection has travelled out of the coupled zone, which serves to prevent spillback. The secondary extension logic for coupled zones is the same as that described above for non-coupled systems. Since coupled intersections cycle together, awarding a secondary extension at the first intersection results in extending all following intersections in the coupled zone. The calculation of the affordable lost time therefore uses the ratio v/c of the critical intersection in the coupled zone.

In a non-critical direction within a coupled zone, on the other hand, the ratio of v/c the local intersection is considered when deciding whether or not to award a secondary extension to a phase. In this case, the secondary extension at one intersection does not necessarily result in secondary extensions in following downstream intersections. The logic for granting a secondary extension for a platoon in a non-critical direction within a coupled zone is the same as before, except for the calculation of ΔC_n, which is instead calculated as follows: if the current intersection is the first or last intersection in a coupled zone, then ΔC_n is the difference between the

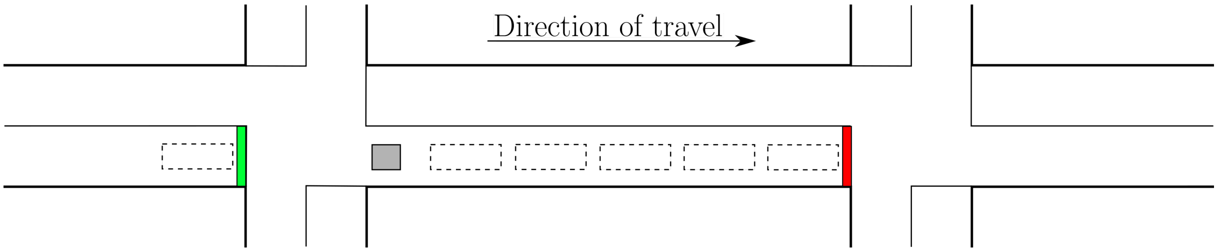


FIGURE 2.8: *A spillback detector upstream from an intersection aimed at preventing spillback into upstream intersection traffic.*

cycle of the critical intersection and the cycle length of the adjacent uncoupled intersection. If the current intersection is somewhere between the first and last intersection of a coupled zone, then ΔC_n is the difference between the cycle length of the local intersection and the cycle length of the critical intersection.

Cesme [12] proposed a few more rules to prevent oversaturation from occurring. The first rule involves the use of a *spillback detector*, as illustrated in Figure 2.8, which detects when spillback is about to occur. When vehicles are queued such that they reach the spillback detector, the green signal of the through phase allowing vehicles to travel downstream is truncated to ensure that vehicles do not block the intersection. In the case where spillback is about to occur in a coupled zone, this rule is only applied when a vehicle stops on the detector and the downstream signal is red. If the signal is green, then the vehicles will be moving very soon due to the short travel time between coupled intersections.

The second rule has to do with so-called *turn pocket spillback*, which occurs when a large portion of approaching vehicles that attempt to turn left at an intersection results in a queue length which extends further than the extent of the left turning lane. An example of this is the situation shown in Figure 2.9. Turn pocket spillback has the potential to block through lanes, significantly decreasing available capacity for a through movement. Two strategies for preventing this kind of spillback are implemented, depending on whether the left turn is leading or lagging, but these strategies are only applied when $v/c > 0.9$. If a left turn is lagging, the spillback detector terminates the green time for the opposing direction only if the minimum green time has been reached, in order for the left-turn movement to start immediately. If a left turn is leading, the spillback detector terminates the current opposing direction and begins a second left-turning phase.

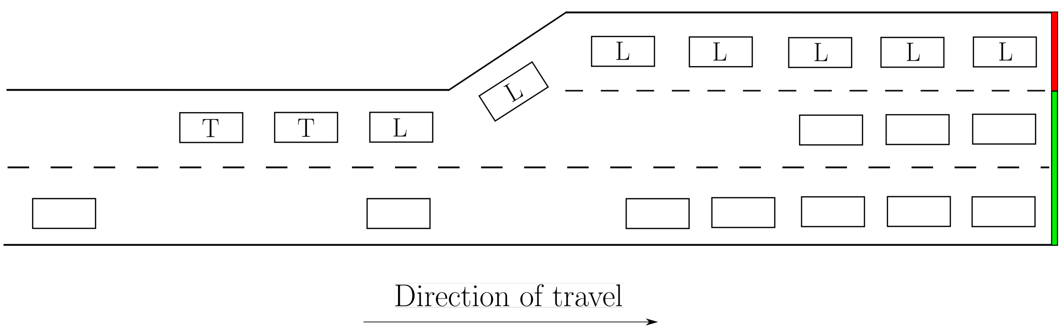


FIGURE 2.9: *Spillback occurring due to a large number of vehicles attempting to turn left (vehicles denoted by L), and preventing through vehicles (denoted by T) from travelling through the intersection.*

A third rule is implemented that deals with oversaturation at coupled intersections. It was stated above that oversaturation at coupled intersections is problematic and so offsets for the critical intersection through phase are determined based on the time taken to travel between intersections. The ideal offset O_{ij} for an intersection i that is upstream from a critical intersection j is given by

$$O_{ij} = Q_{ij} \cdot H_{sat_j} - T_{ij} + Y_j - Y_i, \tag{2.22}$$

where Q_{ij} is the number of vehicles between intersections i and j, and Y_i and Y_j are the lengths of the setup times at intersections i and j, respectively. Furthermore, H_{sat_j} is the saturation flow headway and T_{ij} is the travel time between intersections i and j.

If $O_{ij} > 0$, which is usually the case due to the small distances between coupled intersections, the green arterial through phase at intersection i should begin after that of the critical intersection j. If the previous phase at intersection i has gapped out before the time the through phase is scheduled to start, the previous green signal is held until the scheduled start. If, on the other hand, the previous phase at intersection i has not gapped out by the time of the scheduled start, the previous phase continues until a gap out occurs. This is, however, very unlikely to occur since the critical intersection has the longest cycle length. If $O_{ij} < 0$, then the green arterial through phase of intersection i should begin at an offset from the earliest activation time of intersection j.

The ideal offset O_{kj} for a downstream intersection k (so as to prevent spillback into the critical intersection j) is given by

$$O_{kj} = T_{jk} - Q_{jk} \cdot H_{sat_i} + Y_i - Y_k. \tag{2.23}$$

This offset ensures that the queue at intersection k is cleared in time for the platoon arriving from intersection i. As before, if $O_{kj} > 0$, the green arterial through phase at j should begin after that of the critical intersection k. If $O_{kj} < 0$, then the green arterial through phase of critical intersection j should begin at an offset from the earliest activation time of intersection k.

The last rule deals with the maximum allowed green time a phase can receive during oversaturation. Cesme [12] defined a target maximum cycle length (C_{target}) to be 105s (chosen arbitrarily) which is used to determine maximum green times. For single lane movements, the maximum green time, G_{max}, is given by

$$G_{max} = C_{target} \cdot v/s, \tag{2.24}$$

where v is the volume and s is the saturation flow rate, both of which are updated every cycle. For multilane approaches, the maximum green time is given by

$$G_{max} = C_{target} \cdot v/s \cdot \max(1, X/X_{target}), \tag{2.25}$$

where X is the degree of saturation of the movement (given by $\frac{v}{s}/\frac{g}{C}$, where g and C are the green time and cycle length averaged over the last five cycles). In (2.25), X_{target} is set to 1.0 for the arterial through approach and to 1.1 for all other approaches in order to give a slight priority to the arterial through phases.

This self-organising traffic signal approach was implemented in a simulated environment and compared to coordinated-actuated control optimised by the Synchro software package [79]. A number of different scenarios were tested, including irregularly and closely spaced intersections, as well as uniformly spaced intersections, and it was found that in every instance, the self-organising algorithm performed the best, with 8–14% improvements during undersaturated conditions. Results also suggest that the self-organising algorithm outperforms the coordinated-actuated control even more during oversaturated conditions, with delay reductions of up to 36%.

2.2.2 Three self-organising algorithms proposed by Einhorn

Three self-organising algorithms proposed by Einhorn [16] are described below. These include an algorithm inspired by the theory of inventory control, an algorithm based on the osmosis process and finally a hybrid algorithm, combining the two aforementioned algorithms.

An algorithm inspired by the theory of inventory control

The algorithm described in this section is referred to as the *Inventory Traffic Signal Control Algorithm* (I-TSCA) and it requires the determination of the necessary parameters associated with classical inventory control decisions, namely a reorder point in time as well as an associated reorder quantity. Links can be drawn between traffic control and inventory control in the following way: The product in inventory control can be viewed as the available green time, while vehicles queued at intersections are seen as the customer demand for available stock. Therefore the two quantities that must be determined in this algorithm are the point in time at which signals should change and the length of green time associated with the signal change.

Central to calculating the optimal reorder point and reorder quantity in inventory control, is the calculation of the total cost of the order policy. It is therefore necessary to calculate the total cost associated with traffic control in order to make use of the inventory control analogy in this algorithm. Costs in inventory control are expressed in terms of a currency, but in traffic control it is more convenient to express these costs in terms of vehicle delay. The following four well-known costs associated with classical inventory control models [86] are described, highlighting their corresponding costs in traffic control.

1. The *ordering and setup cost* in inventory theory is the cost of placing an order for stock or the cost of production if the product is produced internally. This kind of cost is fixed for each production run and does not depend on the size of the order or the production run. In traffic control, this cost is considered to be the delay time experienced by queued vehicles at an intersection during the amber and all-red phases that occur just prior to a green signal.

2. The *unit purchasing cost* in inventory theory is the variable cost associated with producing or purchasing a single unit of inventory stock. This cost usually includes the labour cost, overhead costs and raw materials cost associated with producing or purchasing a single unit. In traffic control, however, there is no equivalent cost in terms of vehicle delay.

3. The *holding or carrying cost* in inventory theory is the cost of holding one unit of stock for a single time period. It often includes storage cost, insurance cost, taxes, costs due to spoilage, breakage or theft and most significantly, opportunity cost, which is lost potential interest income by having excessive capital tied up in inventory. The equivalent to this type of cost in traffic control is the delay time experienced by all vehicles currently not receiving service.

4. The *stockout or shortage cost* in inventory control refers to the cost incurred when customer demand is not met on time. This includes a loss of sales and a loss of customer goodwill. In terms of traffic control, this cost can be viewed as the additional delay time experienced by vehicles upon termination of a previous green signal, as they wait for service to resume.

The I-TSCA attempts to minimise the total delay time experienced when vehicles travel through an intersection by calculating the total cost incurred when assigning service to a specific traffic flow phase. The total cost is calculated for each phase, and the phase resulting in the lowest total cost is selected for service and awarded green time. In order to determine the total cost for each phase, the algorithm requires knowledge of the individual distances and speeds of incoming vehicles that are approaching the intersection. Some form of vehicle detection equipment is therefore required in order to calculate the values of these necessary variables. As mentioned, radar detection is assumed to be the mode of detection as it can register the incoming speeds and distances of vehicles from a considerable distance. The I-TSCA makes use of three sets $C_i(t), Q_i(t)$ and $S_i(t)$. First, $C_i(t)$ is the set of all vehicles along an approach lane i at time t. Secondly, $Q_i(t)$ is the set of all queued vehicles at time t, as well as vehicles that will become queued either behind an existing queue or behind a stop line along lane i during a red signal. Lastly, $S_i(t)$ is the set of all stationary, queued vehicles along approach lane i at time t. Therefore, $S_i(t) \subseteq Q_i(t) \subseteq C_i(t)$. The position of a vehicle j in a queue along approach lane i at time t is given by $\epsilon_{ij}(t)$; this value indicates how far the vehicle is from the intersection ahead. A queue position function $\rho_i(t)$ is employed which provides an indication of how far a vehicle queue reaches upstream along lane i at time t. The associated vehicle stopping point in the queue of vehicle j at time t is given by $\mu_j = \rho_i(t)$, as depicted in Figure 2.10. Using these values, the distance between vehicle j at its current position at time t to its future stopping point $\rho_i(t)$ is denoted by $d_{j,\rho_i(t)}$. Once a green signal is displayed at an intersection, queued vehicles are assumed to depart from their positions in approach lane i at a constant rate of η_i vehicles per second, where η_i is the maximum flow rate of Q_i^{max} vehicles per second.

Each cycle of the traffic signals is assumed to consist of a set of B phases as well as B associated setup times. The green time allocated during a phase $m \in B$ is given by χ_m^*, while $\chi_m(t)$ represents the remaining green time of the phase, where $0 \leq \chi_m(t) \leq \chi_m^*$. Similarly, the remaining amount of the total setup time $\mathcal{T}_m^*$ associated with phase m remaining at time t is given by $\mathcal{T}_m(t)$, where $0 \leq \mathcal{T}_m(t) \leq \mathcal{T}_m^*$.

Using these variables, it can be determined whether an approaching vehicle will become queued, and therefore delayed, or travel through the intersection without stopping. If vehicle j is travelling at a speed v_j along approach lane i, the time taken for the vehicle to reach the intersection can be determined so as to predict whether it will become queued or not, and if so, the position and stopping point of the vehicle in the queue may be estimated. If the signal is red, this is achieved by calculating the remaining red time, together with the remaining green time required to clear the queue. If, however, the signal is green, then only the green time taken to clear the queue is required. If the time taken for a vehicle to reach the intersection is more than the remaining green time, it will become queued and is thus added to $Q_i(t)$.

The above decision may be written mathematically as follows: If the current signal is not green and $d_{j,\rho_i(t)}/v_j < \sum_{p\in B\setminus\{m\}}(\chi_p(t) + \mathcal{T}_p(t)) + |Q_i(t)|/\eta_i$, then vehicle j is added to $Q_i(t)$. If the current signal is green, on the other hand, and $d_{j,\rho_i(t)}/v_j < |Q_i(t)|/\eta_i$, then vehicle j is again added to $Q_i(t)$. If the queue is empty, *i.e.* $Q_i(t) = \emptyset$, then vehicle j will become queued in the following two cases: First if the current signal is green and $d_{j,\rho_i(t)}/v_j > \chi_m(t)$ and secondly if the signal is not green and $d_{j,\rho_i(t)}/v_j < \sum_{p\in B\setminus\{m\}}(\chi_p(t) + \mathcal{T}_p(t))$.

The I-TSCA functions by iterating through three main steps. The first of these steps involves the calculation of the required green time $\gamma_i(t)$ for approach lane i to clear all the queued vehicles and vehicles that will become queued before approach lane i receives a green signal, *i.e.* all vehicles in the set $Q_i(t)$. If $Q_i(t) = \emptyset$, then $\gamma_i(t)$ is the time taken for the first vehicle to pass

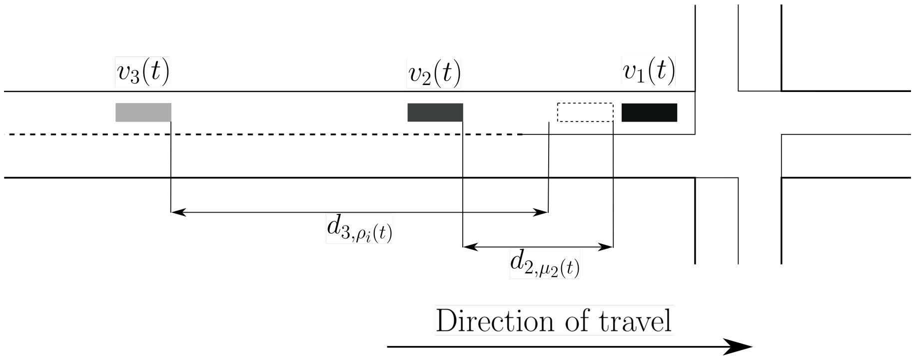


FIGURE 2.10: *A visual representation of the various variables in the I-TSCA by Einhorn [16]. Assuming the traffic signal ahead of the vehicles is not green and* $d_{3,\mu_3(t)}/v_3(t) < \sum_{p\in B\setminus\{m\}}(\chi_p(t)+\mathcal{T}_p(t))+\frac{Q_i(t)}{\eta_i}$, *then* $C_i(t)=\{v_1(t);v_2(t);v_3(t)\}$, $Q_i(t)=\{v_1(t);v_2(t);v_3(t)\}$ *and* $S_i(t)=\{v_1(t)\}$.

through the intersection. For an approach lane i, the required green time is given by

$$\gamma_i(t)=\begin{cases} d_{1,\rho_i(t)}(t)/v_1(t), & \text{if } Q_i(t)=\emptyset \text{ and } C_i(t)\neq\emptyset,\\ |Q_i(t)|/\eta_i, & \text{if } Q_i(t)\neq\emptyset,\\ \infty, & \text{if } C_i(t)=\emptyset. \end{cases} \tag{2.26}$$

The first condition in (2.26) is the case in which there are no queued vehicles, but there is at least one vehicle approaching the intersection along approach lane i, and the required green time is the time it will take for that first vehicle to pass through the intersection. The purpose of this condition is to ensure that intersection usage is maximised. The second condition in (2.26) is the case in which there is at least one queued vehicle, and so the time required to clear the queue is the number of vehicles in the queue, divided by the number of vehicles that depart from the queue per second. Suppose U_m is the set of all approach lanes receiving service during phase m and $\gamma_i(t)$ is calculated for each approach lane i. The minimum required green time over all the lanes of phase m is chosen, *i.e.* $\gamma_m(t)=\min_{i\in U_m}\gamma_i(t)$. The purpose of this condition is to ensure that if one lane has a very large required green time, it does not prevent other lanes requiring shorter green times from receiving service. The third condition in (2.26) assigns a value of infinity to $\gamma_i(t)$ due to the reasoning that if there are no approaching vehicles along approach lane i, then approach lane i must not receive service.

The second step of the I-TSCA involves the calculation of vehicle delays associated with awarding service to a certain phase. Since a vehicle will experience a delay if it becomes queued, it is necessary to predict the length of time over which the vehicle will be queued. Because this value depends on whether approach lane i is receiving service or not, a binary parameter $\kappa_i(t)$ is introduced, where $\kappa_i(t)=0$ indicates no service, while $\kappa_i(t)=1$ indicates that approach lane i is receiving service. Therefore, the delay time experienced by a vehicle j along approach lane i during phase m is

$$\phi_{ij}^m(t)=\begin{cases} \mathcal{T}_m(t)+\frac{\epsilon_{ij}(t)}{\eta_i}-\frac{d_{j,\mu_j(t)}(t)}{v_j(t)} & \text{if } \kappa_i(t)=1,\\ \sum_{p\in B}\mathcal{T}_p(t)+\sum_{p\in B\setminus\{m\}}\chi_p(t)-\frac{d_{j,\mu_j(t)}(t)}{v_j(t)}+\frac{\epsilon_{ij}(t)}{\eta_i}, & \text{if } \kappa_i(t)=0. \end{cases} \tag{2.27}$$

If vehicle j is not queued along approach lane i, the vehicle will not experience a delay and so $\phi_{ij}^m(t)=0$. The first condition in (2.27) represents the various times associated with receiving service. The sum of the first two terms represents the time taken for the intersection to be cleared, while the last term represents the time taken for the vehicle to reach the intersection.

The second condition in (2.27) contains four terms, the first being the sum of the remaining setup times of various phases. The second term represents the remaining green time of the phases preceding the green signal. The third and fourth terms are the same as described in the first line of (2.27).

Algorithm 2.5: The inventory traffic signal control algorithm.

Input : The positions and speeds of all vehicles approaching the intersection at each time step.

Output: A signal phase switching decision for each time step.

for *each time step* t **do**
    for *each approach lane* i *of the intersection* **do**
        if $|Q_i(t)| = 0$ **and** $|C_i(t)| > 0$ **then**
            $\gamma_i(t) \leftarrow d_{1,\rho_i(t)}(t)/v_1(t)$;
        else if $0 < |Q_i(t)| < \infty$ **and** $|C_i(t)| > 0$ **then**
            $\gamma_i(t) \leftarrow |Q_i(t)|/\eta_i$;
        else if $|C_i(t)| = 0$ **then**
            $\gamma_i(t) \leftarrow \infty$;
        for *each vehicle* j *on approach lane* i **do**
            if $\kappa_i(t) = 1$ **then**
                $\phi_{ij}^m(t) \leftarrow \tau_m(t) + \frac{\epsilon_{ij}(t)}{\eta_i} - \frac{d_{j,\mu_j}(t)}{v_j(t)}$;
            else if $\kappa_i(t) = 0$ **then**
                $\phi_{ij}^m(t) \leftarrow \sum_{p\in B} \tau_p(t) + \sum_{p\in B\setminus\{m\}} \chi_p(t) + \frac{\epsilon_{ij}(t)}{\eta_i} - \frac{d_{j,\mu_j}(t)}{v_j(t)}$;
    for *each signal phase* m **do**
        $\Gamma_m(t) \leftarrow min_{i\in U_m}\gamma_i(t)$;
        $\Phi_m(t) \leftarrow \sum_{i\in I}\sum_{j\in C_i(t)} \phi_{ij}^m(t)$;
        if $\Phi_m(t) < \Phi_{m'}(t)$ **and** $\chi_m(t) = 0$ **then**
            $\kappa_i(t) \leftarrow 1 \forall i \in U_m$;
            $\chi_m(t) \leftarrow \Gamma_m(t)$;

Let I be the set of all the approach lanes to the intersection. The total cost $\Phi_m(t)$ of assigning phase m service is determined by summing the delay terms incurred by all vehicles across the approach lanes in I. Therefore, $\Phi_m(t) = \sum_{i\in I}\sum_{j\in C_i(t)} \phi_{ij}^m(t)$. The different costs associated with the delay time are as follows: The setup cost is calculated by summing the setup time $\mathcal{T}_m(t)$ for all delayed vehicles. The holding cost is calculated by summing all delay times incurred by vehicles that are not currently receiving service, and the stock-out cost is the sum of the delay times experienced by vehicles whose service is terminated before they can travel through the intersection.

The third and final step of the I-TSCA is assigning a service time of length $\Gamma_m(t)$ to a specific phase m such that $\Phi_m(t)$ is a minimum. Once service has been assigned, the I-TSCA continues to recalculate the required green time and total cost, and assigns green time once the current service time $\Gamma_m(t)$ has elapsed. It is possible that after the service time of length $\Gamma_m(t)$ elapses at t', phase m is again selected for service, in which case service is not terminated, but is extended for another $\Gamma_m(t')$ time units.

A pseudo code description of the I-TSCA for this algorithm is given in Algorithm 2.7. In lines 3 to 8, the green time required to clear the current queue is determined (the first step), while in lines 9 to 13 the delay time for vehicles is calculated (the second step). In lines 15 to 17, the required green time is minimised and the associated total vehicle delay is calculated. Finally, in lines 18 to 20, service is awarded to the approach resulting in the lowest total vehicle delay, given that the remaining green time for the previous phase is zero.

An algorithm inspired by the chemical process of osmosis

Another self-organising traffic signal control algorithm proposed by Einhorn [16] is known as the *Osmosis Traffic Signal Control Algorithm* and is abbreviated here as O-TSCA. Osmosis is a chemical process whereby solvent molecules move from a solution containing a low solute concentration, through a semipermeable membrane, to a solution containing a higher solute concentration. The solution containing the lower solute concentration has a higher *osmotic pressure*, while the solution containing the higher solute concentration has a lower osmotic pressure. The difference between these pressures defines the *osmotic gradient*. Over time, as these concentrations become equal on both sides of the membrane, the osmotic gradient tends towards zero. This process is depicted in Figure 2.11.

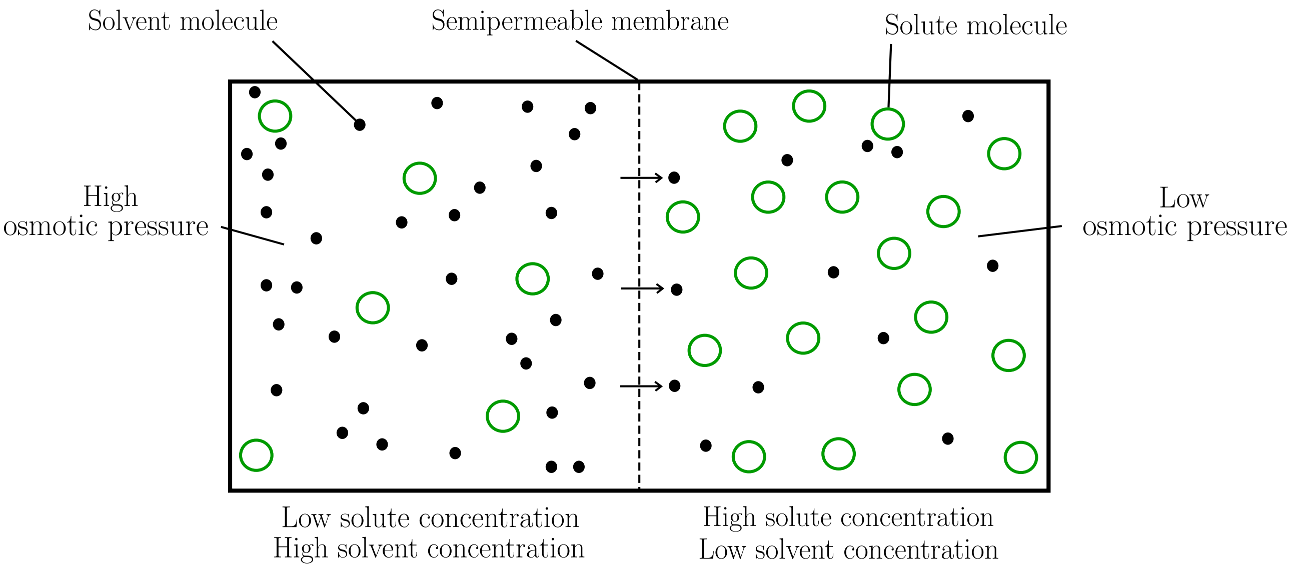


FIGURE 2.11: *An illustration of the process of osmosis.*

A number of analogies may be drawn between the various components of the osmotic process and the basic elements of traffic control. The O-TSCA likens the solvent molecules to vehicles travelling through an intersection, which is analogous to the semipermeable membrane through which the solvent molecules (vehicles) pass. The solute molecules further correspond to the empty space on the opposite side of the intersection. Vehicles approaching an intersection may therefore be regarded as exerting a *push pressure* on the intersection so that the vehicles are pushed through the intersection in a manner similar to how solvent molecules are pushed through the membrane as a result of the osmotic gradient. The empty space along the roadway just after the intersection may similarly be seen as exerting a *pull pressure* on the intersection so that vehicles are pulled through the intersection analogously to how solvent molecules are pulled through the semipermeable membrane as a result of the osmotic gradient.

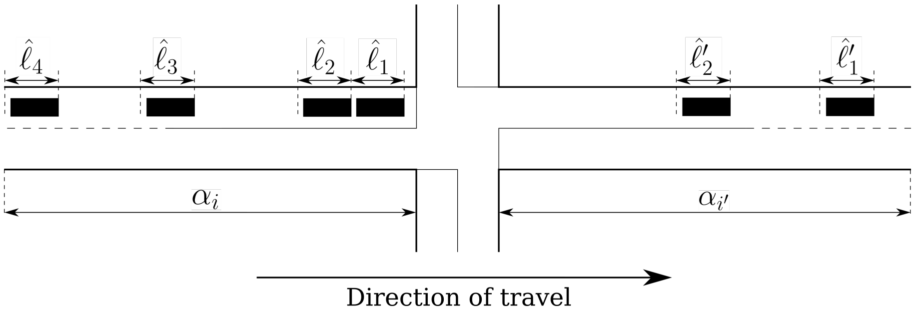


FIGURE 2.12: *The demand along approach lane i and the corresponding exit lane i' is given by $\delta_i(t) = \sum_{j=1}^{4} \hat{\ell}_j$ and $\delta_{i'}(t) = \sum_{j=1}^{2} \hat{\ell}'_j$, respectively. The availability associated with lane i is $\omega_i(t) = \alpha_{i'} - \sum_{j=1}^{2} \hat{\ell}'_j$, and finally the pressure associated with lane i is $\pi_i(t) = \sum_{j=1}^{4} \hat{\ell}_j + \alpha_{i'} - \sum_{j=1}^{2} \hat{\ell}'_j$.*

Unlike the I-TSCA, the O-TSCA does not take vehicle delay into account at all when determining which phase to award service to. The O-TSCA instead functions by calculating the demand and availability associated with approach lane i to the intersection and the corresponding exit lane i' from the intersection, respectively. Similarly to the I-TSCA, the O-TSCA assumes the use of appropriate radar detection equipment at the intersection, in order for individual vehicle characteristics, such as speed and distance from the intersection, to be recorded.

Let α_i be the length of approach lane i and let $\hat{\ell}_j$ denote the effective length of vehicle j, which is determined by summing the physical length of the vehicle and a safety gap that is always maintained between stationary vehicles. The demand $\delta_i(t)$ associated with approach lane i measures the amount of space occupied along the lane and is calculated by determining the number of approaching vehicles along lane i and summing over their effective lengths, *i.e.* $\delta_i(t) = \sum_{j \in C_i(t)} \hat{\ell}_j$, where $C_i(t)$ is the set of all vehicles along approach lane i at time t. The availability $\omega_i(t)$ associated with approach lane i represents the effective space along the corresponding exit lane i' that is available for vehicles to occupy, *i.e.* $\omega_i(t) = \alpha_{i'} - \delta_{i'}(t)$. The pressure $\pi_i(t)$ exerted by approach lane i at the intersection at time t is expressed as the sum of demand and availability associated with lane i at time t. In other words,

$$\pi_i(t) = \delta_i(t) + \omega_i(t). \tag{2.28}$$

The following statements can be made about (2.28): First, a high demand and low availability or a low demand and a high availability will both result in medium pressures. Secondly, a high demand and high availability for a particular lane will lead to a high pressure, and thirdly, a low demand and low availability will result in a low pressure.

The O-TSCA also calculates a throughput for each lane, which is the sum of the effective lengths of the vehicles that have travelled through a particular intersection. The throughput, denoted by $\theta_i(t)$, provides an indication of the total space occupied by these vehicles.

In order to determine the pressure of a phase m over all the approach lanes that receive service during the phase, the pressures $\pi_i(t)$, are summed over all the approach lanes receiving service during phase m. The total pressure for phase m is therefore given by $\Pi_m(t) = \sum_{i \in U_m} \pi_i(t)$, where U_m is the set of all approach lanes receiving service during phase m. The total pressure for each phase is calculated and the phase with the largest pressure is awarded service. If phase m is selected for service at time t^*, the following two variables are stored by the O-TSCA: First, the *demand*, given by $\Delta_m = \sum_{i \in U_m} \delta_i(t^*)$, and secondly, the *availability*, given by $\Omega_m = \sum_{i \in U_m} \omega_i(t^*)$. Phase m continues to receive service until

$$\sum_{i \in U_m} \theta_i(t) \geq \Delta_m, \tag{2.29}$$

or

$$\sum_{i \in U_m} \theta_i(t) \geq \Omega_m. \tag{2.30}$$

Inequality (2.29) requires that phase m receives service until the total throughput of lanes served during phase m is equal to or larger than the total demand of all lanes served during phase m. In other words, it ensures that all vehicles initially requiring service, are served. Inequality (2.30) requires that phase m receives service until the total throughput of the lanes served during phase m is equal to or larger than the total availability of the lanes served during phase m. In other words, this constraint ensures that the space required by vehicles receiving service along approach lane i does not exceed the amount of available space along the corresponding exit lane i'.

Once condition (2.29) or condition (2.30) holds, the O-TSCA recalculates the pressures for each phase $m \in U_m$ and once again awards service to the phase with the largest pressure, which may or may not be the same phase currently receiving service. If, however, service is awarded to a different phase, the throughput $\theta_i(t)$ is set to zero for all $i \in U_m$.

Algorithm 2.6: The osmosis traffic signal control algorithm.

Input : The physical lengths of all vehicles approaching the intersection at each time step.
Output: A signal phase switching decision for each time step.

1 **for** *each time step t* **do**
2   **for** *each approach lane i of the intersection* **do**
3         $\delta_i(t) \leftarrow \sum_{j \in C_i(t)} \hat{\ell}_j$;
4         $\omega_i(t) \leftarrow \alpha_{i'} - \delta_{i'}(t)$;
5         $\pi_i(t) \leftarrow \delta_i(t) + \omega_i(t)$;
6   **for** *each signal phase m* **do**
7         $\Pi_m(t) = \sum_{i \in U_m} \pi_i(t)$;
8   **if** $\sum_{i \in U_m} \theta_i(t) \geq \Delta_m$ **or** $\sum_{i \in U_m} \theta_i(t) \geq \Omega_m$ **then**
9   **if** $\Pi_m(t) > \Pi_{m'}(t)$ **then**
10                 $t \leftarrow t^*$;
11                 $\Delta_m \leftarrow \sum_{i \in U_m} \delta_i(t^*)$;
12                 $\Omega_m \leftarrow \sum_{i \in U_m} \omega_i(t^*)$;
13                 $\theta_i(t) \leftarrow 0 \forall j \in U_{m'}$:

A pseudo code description of the O-TSCA is given in Algorithm 2.6. In lines 3 to 5, the demand, availability and pressure for each lane i are calculated, while the pressures for each phase are summed in line 7. In line 8, the algorithm checks whether the throughput of phase m is equal to or larger than the demand or availability. If this is the case, the algorithm then checks whether phase m exerts a larger pressure than competing phase m' (line 9). If this is the case, the algorithm switches signals, assigning new demand and availability values in lines 11 and 12 and terminating by resetting the throughput to zero in line 13.

A hybrid algorithm

It was found by Einhorn [16] that the I-TSCA performed most efficiently under light traffic conditions due to the fast switching propensity of the algorithm, while under heavy traffic conditions the signals switched too frequently, often separating platoons of vehicles through the intersection. The O-TSCA, on the other hand, appears to work better under heavy traffic conditions due to the sufficiently long green times associated with the algorithm, while under light traffic conditions the O-TSCA may be less effective, awarding green times that are longer than necessary.

The third and final algorithm designed by Einhorn [16] is referred to as the *Hybrid* algorithm. The Hybrid algorithm attempts to capitalise on the advantages of both the I-TSCA and the O-TSCA by implementing a control strategy which is essentially a combination of the I-TSCA and the O-TSCA. The Hybrid algorithm functions by executing both the I-TSCA and O-TSCA concurrently, together with an *intersection utilisation maximisation supervisory mechanism* (IUMSM), in order to ensure that the intersection is not under-utilised, *i.e.* green times are not too long or too short.

The Hybrid algorithm achieves this by making use of a binary variable, known as the *proximity* $\xi_i(t)$, associated with approach lane i as well as a constant, safe following distance from the intersection, denoted by e. In particular,

$$\xi_i(t) = \begin{cases} 1, & \text{if the closest vehicle in lane } i \text{ is within a distance } e \text{ from intersection,} \\ 0, & \text{otherwise.} \end{cases} \tag{2.31}$$

The proximity variable is calculated for each phase m and these values are summed together. That is, $\Xi_m(t) = \sum_{i \in U_m} \xi_i(t)$. The IUMSM works as follows. If the I-TSCA requests a signal change, then the IUMSM determines whether or not there is an approaching vehicle j along approach lane i currently receiving service, within a safety distance e from the intersection. The IUMSM therefore ascertains whether $\Xi_i(t) = 0$ or $\Xi_i(t) = 1$. If $\Xi = 0$, then a signal change occurs. If, on the other hand, $\Xi_i(t) = 1$, then the current phase continues until either there are no longer any vehicles along approach lane i within a distance e from the intersection or until the O-TSCA also indicates that a signal change is necessary. If a vehicle along approach lane i that is not currently receiving service, reaches the intersection at time t, where t is less than the duration of an amber and all-red phase, and there is no vehicle along the approach lane receiving service, the IUMSM will request a signal change if neither the I-TSCA nor the O-TSCA has done so.

Einhorn [16] found that the Hybrid algorithm performed better than both the I-TSCA and O-TSCA in terms of reducing vehicle delay under light traffic conditions. Under heavy traffic conditions, however, Hybrid was ranked the second most effective out of six different algorithms considered, outperformed only by O-TSCA.

2.3 Chapter summary

Three earlier, self-organising traffic signal control algorithms were described in §2.1. Seminal algorithms by Gershenson and Rosenblueth [22], and by Lämmer and Helbing [46] were described in §2.1.1 and §2.1.2, respectively, while an algorithm by Xie *et al.* [87] was described in §2.1.3. In §2.2 a number of more recently proposed algorithms by Cesme [12] and Einhorn [16] were presented. The algorithm of Cesme [12] focused on traffic control for arterial roads, while three algorithms were proposed by Einhorn [16] for a general context: an algorithm inspired by the theory of inventory control (described in §2.2.2), an algorithm inspired the chemical process of osmosis (described in §2.2.3), and an algorithm that combines both the aforementioned strategies (described in §2.2.4).

The Algorithm by Gershenson and Rosenblueth [22], as well as the algorithm by Cesme [12], is rule-based. Gershenson and Rosenblueth organise their algorithm in such a manner that it

Algorithm 2.7: The hybrid traffic signal control algorithm.

Input : The positions, speeds and physical lengths of all vehicles approaching the intersection at each time step.

Output: A signal phase switching decision for each time step.

for *each time step t* **do**
    for *each approach lane i of the intersection* **do**
        $\delta_i(t) \leftarrow \sum_{j \in C_i(t)} \hat{\ell}_j$;
        $\omega_i(t) \leftarrow \alpha_{i'} - \delta_{i'}(t)$;
        $\pi_i(t) \leftarrow \delta_i(t) + \omega_i(t)$;
        if $|Q_i(t)| = 0$ **and** $|C_i(t)| > 0$ **then**
            $\gamma_i(t) \leftarrow d_{1,\rho_i(t)}(t)/v_1(t)$;
        else if $0 < |Q_i(t)| < \infty$ **and** $|C_i(t)| > 0$ **then**
            $\gamma_i(t) \leftarrow |Q_i(t)|/\eta_i$;
        else if $|C_i(t)| = 0$ **then**
            $\gamma_i(t) \leftarrow \infty$;
        for *each vehicle j on approach lane i* **do**
            if $\kappa_i(t) = 1$ **then**
                $\phi_{ij}^m(t) \leftarrow \tau_m(t) + \frac{\epsilon_{ij}(t)}{\eta_i} - \frac{d_{j,\mu_j}(t)}{v_j(t)}$;
            else if $\kappa_i(t) = 0$ **then**
                $\phi_{ij}^m(t) \leftarrow \sum_{p \in B} \tau_p(t) + \sum_{p \in B \setminus \{m\}} \chi_p(t) + \frac{\epsilon_{ij}(t)}{\eta_i} - \frac{d_{j,\mu_j}(t)}{v_j(t)}$;
    for *each signal phase m* **do**
        $\Gamma_m(t) \leftarrow min_{i \in U_m} \gamma_i(t)$;
        $\Phi_m(t) \leftarrow \sum_{i \in I} \sum_{j \in C_i(t)} \phi_{ij}^m(t)$;
        if $\Phi_m(t) < \Phi_{m'}(t)$ **and** $\chi_m(t) = 0$ **then**
            $\kappa_i(t) \leftarrow 1 \ \forall i \in U_m$;
            $\chi_m(t) \leftarrow \Gamma_m(t)$;
        $\Pi_m(t) \leftarrow \sum_{i \in U_m} \pi_i(t)$;
        if $\Pi_m(t) > \Pi_{m'}(t)$ **then**
            $t \leftarrow t^*$;
            $\Delta_m \leftarrow \sum_{i \in U_m} \delta_i(t^*)$;
            $\Omega_m \leftarrow \sum_{i \in U_m} \omega_i(t^*)$;
            $\theta_i(t) \leftarrow 0 \ \forall j \in U_{m'}$:

is very simple to understand, while the algorithm of Cesme [12] consists of a large number of rules that lack structure in their presentation. The algorithm by Xie *et al.* [87] has a number of parameter values that have to be selected by the user (threshold values and detector location) and assumes one-way traffic only. Both the algorithm by Xie *et al.* [87] and the algorithm by Cesme [12] assume stop line detectors and upstream detectors as the mode of detection and make use of various detector lengths in the algorithmic calculations, while the remaining five algorithms make use of more descriptive data such as continuous updates of vehicle distances from intersections and flow rates, which cannot be obtained as accurately from the use of vehicle inductive loop detectors. Due to these reasons, these two algorithms are not implemented in this dissertation.

CHAPTER 3

Simulation modelling

Contents

This chapter serves as a brief introduction to the field of simulation, starting with a general description of simulation as well as some important simulation modelling concepts in §3.1. This is followed in §3.2 by the descriptions of the four main types of simulation models found in the literature. Some advantages and disadvantages of simulation modelling are mentioned in §3.3 and the twelve typical steps followed in a simulation study are briefly discussed in §3.4. Various methods of simulation model verification and validation are reviewed in §3.5, and this is followed by a description of three different fundamental types of traffic simulation models and a number of different traffic simulation software packages in §3.6. The chapter closes with a brief summary in §3.7.

3.1 Modelling concepts

Simulation modelling essentially involves the creation of a hypothetical system which mimics the behaviour of a real system. According to Banks [6], simulation entails the generation of an "artificial history" which may be used to draw conclusions about the actual underlying system. It is a problem solving methodology that allows complex problems, which cannot be modelled analytically, to be analysed numerically by means of a computer [6, 49]. Simulation modelling is a powerful tool that can be used to improve one's understanding of an existing system as well as serve as a method for comparing a number of different control policies applicable to the system under consideration [53, 71].

While there are various types of simulation paradigms, there are a few important components that make up the basic structure of most simulation models. The *system, model, events, entities, attributes, activities, resources* and *system state variables* are common modelling concepts underlying a simulation model. These concepts are described briefly in this section.

A *system* may be defined as a set of principles or processes consisting of an organised group of *entities* whose interactions strive towards some common goal [23, 71, 86]. A *model* is a representation of a real *system* together with a set of assumptions made in order to predict how these principles or processes will perform if physically implemented [6, 49, 71]. The state of a system is influenced by *events* which are occurrences that have the potential to change *system state variables* [6, 49] which, in turn, changes the state of the system.

An *entity* is an object, such as a person or machine, in a system that is either dynamic and moves through the system (*e.g.* customers) or static and remains stationary within the system (*e.g.* a cashier) [6, 71]. Entities have certain unique characteristics (*e.g.* names or priorities) which are known as *attributes* and are used to describe the behaviour of the entities. Entities are the actual cause of changes in the system since when they interact with *activities*, the result is the occurrence of events [6, 37, 71]. *Activities* are processes that occur in a simulation, the three most common being delays, queues and logic [37]. A *resource* is a special kind of entity that has a restricted capacity and serves other dynamic entities in the system. Resources can be bank tellers, machines or even traffic intersections, and may assume one of a number of states, such as idle, busy, blocked and failed.

The *state* of a system is the description of the current system status at a certain instant in time through the use of variables [6, 49, 86]. These variables are referred to as *system state variables* and they vary with respect to the type of process modelled in the simulation. In the case of a bank hall simulation, for example, the system state variables may be the numbers of busy and idle tellers, the total number of customers in the bank hall and the arrival times of customers.

Simulation models may be classified as either *static* or *dynamic*, and as either *stochastic* or *deterministic*, while events in a simulation model may be either *discrete* or *continuous*. A *static* model is a model that is independent of time, and therefore represents a system at a certain instant in time, rather than a system that evolves over time [6, 23]. This kind of simulation model is commonly referred to as a *Monte Carlo* simulation [86] and an example of this kind of model is a simulation that models the outcome of rolling a die. *Dynamic* models, on the other hand, are models that change with respect to time [6, 23, 37, 53, 68]. In this case, a so-called simulation clock is incorporated into the model which simulates the seconds or hours or days that have elapsed since initialising an instance of the model. The processes involved in the manufacturing of a product is an example of where a dynamic simulation model may be applicable.

A model containing *stochastic* processes is a model that makes use of randomness [53, 71]. The output of the model is different each time the model is run, and variables may be defined by means of probability distributions [68]. An example of a stochastic simulation is modelling the movement of customers in a bank, where customer inter-arrival and service times are defined by means of probability distribution functions. *Deterministic* models, on the other hand, contain no random variables and assume certainty in every aspect [23, 53]. Results obtained from these models are therefore entirely independent of probability. An example of this is a pricing structure, whose model is solely based on known average values.

Dynamic models change over time due to the occurrence of events and these changes can either occur discretely or continuously. A *discrete* event is an action that occurs within the simulation at distinct points in time, such that the state of the system changes only at these discrete times [23, 53, 86]. *Continuous* events, on the other hand, are actions that occur continuously, and not just at certain points time [86]. An example of a continuous event is the simulated change of temperature of an object.

3.2 Types of simulation modelling paradigms

There are four major types of simulation modelling paradigms that can be adopted, based on the nature of the simulation and the required level of abstraction necessary.

3.2.1 Agent-based modelling

An *agent-based model* may be defined as a system containing a collection of self-governing entities referred to as agents, that have unique characteristics defining them and their behaviour [8]. Each agent is responsible for making individual decisions based on a set of rules that it must follow, which results in certain collective behaviours that are exhibited within the system. People, vehicles, companies and products are all examples of possible agents in a system, while main drivers, reactions and history are examples of possible collective behaviours emerging from the actions of agents. The system is decentralised as model behaviour is defined at a microscopic level, while global system behaviour emerges due to the many interactions between individual agents. Because of these properties, agent-based modelling is the simulation modelling paradigm adopted in this dissertation.

3.2.2 Discrete-event modelling

Discrete-event modelling is a modelling approach in which events occur sequentially at discrete points in time. Since events occur only at specific times, the state of the system does not change between consecutive events; it only does so at the instants at which the events occur. This kind of modelling approach can describe the system according to a flowchart of model logic, where entities "travel" through various blocks in the flowchart. A common example of a discrete-event simulation is a model of a bank where customers arrive and queue to be served by tellers.

3.2.3 System dynamics modelling

System dynamics modelling is a modelling paradigm that is used to understand the structure and non-linear, complex behaviour of dynamic systems [74]. Unlike agent-based modelling, however, such an approach is more abstract and assumes aggregation of the objects being modelled such that objects are represented by their quantity, losing any individual properties they may have had [9, 75]. Processes are represented by stocks (people, money, materials), flows between these stocks as well as by information affecting the value of these stocks [9].

3.2.4 Dynamical systems modelling

Dynamical systems modelling is the ancestor paradigm of system dynamics modelling and is applied in the conventional design process of projects in technical engineering disciplines [9]. According to this modelling paradigm, the system is described by a number of differential equations relating various system state variables [2, 59]. In contrast to system dynamics, the integrated state variables have physical meaning (*e.g.* location, speed or length) and there is no aggregation of objects [9, 2].

3.3 Benefits and shortcomings of simulation modelling

Simulation is a widely used and very powerful modelling tool that is commonly applied in the investigation of complicated systems which, if understood and used correctly, can offer a variety of benefits when designing a system model [6, 49, 53, 68].

One advantage of adopting a simulation modelling paradigm is that it allows for the testing of additions or modifications to a system, without committing any actual capital to these alterations [6, 53, 68, 71]. This allows for the efficiency of a system to be determined before physical implementation, therefore enabling a design project to be terminated before resources are committed (if it appears that the simulated model does not behave as expected). Simulation modelling also offers the convenience of time speed variation [6, 49, 53, 71]. The model time over which the simulation evolves can be slowed down dramatically in order to observe interactions occurring during short time intervals, or can be increased in order to arrive at a certain point of interest in the simulation sooner. Watching simulation visualisations can also help the decision maker understand why certain phenomena occur, by recreating scenarios that display a particular kind of behaviour [6, 68].

One of the most considerable advantages of simulation modelling is that once such a model has been built and validated, it can be used to ascertain the effects of various modifications to the underlying system [6]. The effects of these modifications can be observed in the simulation environment rather than in the actual system, reducing the risk of running into unforeseen problems. Simulation also aids the user in identifying certain microscopic problems in a process that may be difficult to spot in the real system [6]. This may occur when multiple interactions occur in a complex system close to simultaneously, while in a simulation model of the system the decision maker may slow down the simulation time and observe a number of these interactions.

When a simulation model is built from scratch, it provides the modeller with a deeper understanding of the process being modelled, allowing for better interpretation of the results and the formulation of more valid feedback in respect of the underlying system dynamics than someone who is not familiar with the model [6, 53]. Another benefit of simulation modelling is that it is often a good investment [6]. Typically, the cost of a simulation study is less than 1% of the total implementation cost of a system, which is comparatively low [6].

Simulation models may also be used in the training of new employees [6]. It can assist employees in learning and improving their performance before they are placed in the actual workplace where their presence may be disruptive and they have the potential to cause financial losses if not fully trained.

In cases where other models are too complicated to be analysed analytically, simulation is a tool that offers a practical alternative [49]. Another advantage of simulation is that once a valid and robust simulation model has been developed, the time in which the real system is subsequently built, may be substantially reduced. Simulation modelling also allows for a number of different designs to be compared in order to determine which design is best suited for implementation [49].

Simulation is, however, not without drawbacks. A disadvantage associated with simulation modelling is that the individual in charge of creating the simulation model needs to be trained in the use of a specific simulation software [6, 23, 71]. This can cost a significant amount of money and often is not a quick process as competence in simulation is typically learnt gradually over time and through extensive experience.

Creation of a simulation model generally also requires a number of assumptions to be made. If an assumption is incorrect, or not documented, then certain conclusions drawn from the simulation results may not be valid. It may also sometimes be difficult to interpret results obtained from a simulation study [6]. Since simulations are based on a certain degree of randomness, it may be difficult to determine whether a particular result occurred due to randomness or due to actual prolonged relationships between elements in the system [6]. Another drawback of simulation modelling is that the model creation process can be time consuming and costly, since simulation packages that are not available as open source generally have very expensive licenses.

3.4 Typical steps in a simulation study

In order to create an effective simulation model, there are a number of steps that should be followed in any simulation study. These steps are summarised in the flowchart in Figure 3.1 and are described briefly in this section.

1. *Problem identification and formulation.* Every simulation study begins with a problem statement clearly defining the problem that requires solution, which must be well understood by the individual or team who will be performing the study [5, 49]. This is important as the purpose of the study must be clear in order to ensure a successful model. It may, however, happen that the problem statement is reformulated further into the simulation study process due to unexpected findings discovered along the way or the gaining of a better understanding of the underlying system being modelled.

2. *Set objectives and establish a project plan.* The objectives of the study are defined in order to determine which questions have to be answered by the end of the study [5, 49, 71]. The project plan includes the different scenarios of the model that will be investigated as well as the estimated time and cost of completing the simulation study. The project plan may also contain the required staff and equipment, the output expected at different stages of the simulation modelling process as well as any other requirements of the client for whom the simulation study is performed.

3. *Conceptualise the model.* The mathematical relationships and interactions between different components in the system are identified and processes based on these are defined in order to develop a theoretic model of the underlying system [5, 68, 71]. It is recommended that the conceptual model be moderately simple, as a complicated, very detailed model may be computationally expensive and take longer to build [5, 49].

4. *Data collection.* Data required to populate the model are collected from the client if available, which is useful for defining realistic probability distribution functions for the model [5, 49, 53, 68, 71].

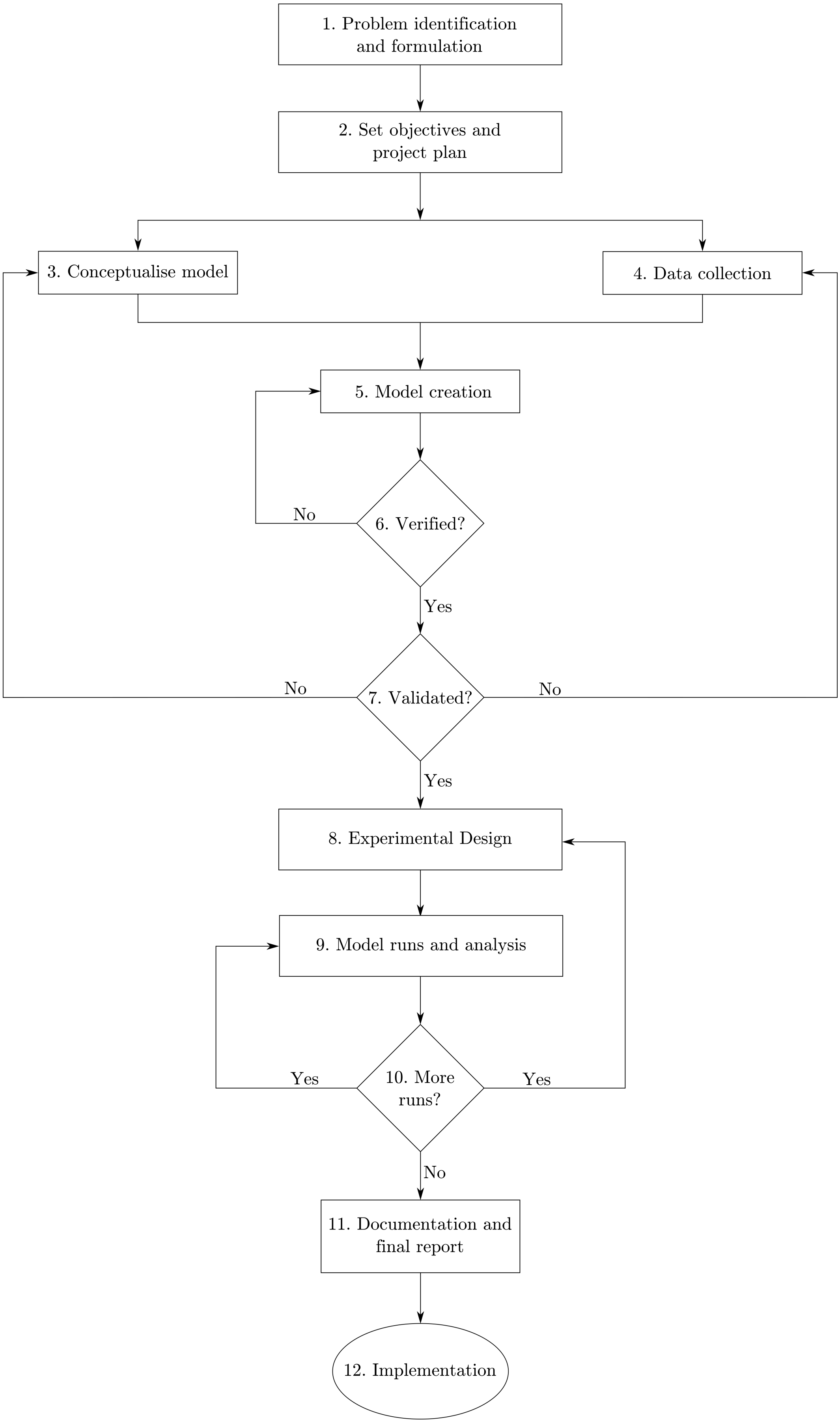


FIGURE 3.1: *A flowchart representing the steps in a simulation study [6].*

5. *Model creation.* The model is translated into the appropriate computer simulation software by programming the logic formulated in step 3 in an appropriate simulation modelling language [5, 68, 71].

6. *Model verification.* The verification process involves determining whether the simulation model accurately represents the model that was formulated and built [5, 71]. It is a process that should be performed continuously throughout the model building process, often through the process of debugging [68].

7. *Model validation.* Validation is the process of determining whether the simulation model accurately represents the real underlying system. A good way of validating a simulation model involves comparing the outputs of the simulation model with actual outputs taken from the real system, although this is not always possible [5, 49, 68, 71].

8. *Experimental design.* A number of experiments are designed that are necessary to yield the required output from the model [49, 68, 71]. This includes the initialisation of the model, determining the length of a single simulation run as well as the total number of simulation runs that have to be performed [5, 49].

9. *Model runs and analysis.* The simulated model is executed over the course of various experiments and performance data are produced. Statistical measures are used to analyse these data, including the determination of confidence intervals and performing a sensitivity analysis [5, 49, 68].

10. *Additional runs.* After the initial set of model runs have been completed and have been analysed, it is decided whether or not additional runs are required for the performance of further experiments [5].

11. *Model documentation and report.* Proper documentation of the simulation code is important, especially if the model will be used by others who are not familiar with the implementation of the model or if modifications to the model are expected to be carried out in the future [5, 49, 68, 71]. The final report should be written succinctly and include the assumptions made throughout the modelling process, the different experiments carried out, the analysis of the results obtained and the final conclusions of the simulation study.

12. *Model implementation.* The client is ultimately the one making the decision regarding whether or not to implement the model in a real system. The results and success of the earlier steps are central to the outcome of this final decision [5, 68].

3.5 Verification and validation of a simulation model

Verification and validation of a simulation model is typically time consuming, but is important to ensure the output of a successful model [71]. The verification and validation processes (illustrated in Figure 3.2) are both aimed at producing a suitable model. Verification is concerned with building the model correctly, whereas validation is concerned with building the correct model [4, 71].

3.5.1 Verification of a simulation model

Verification of a simulation model involves determining whether or not the model behaves as is expected [71], *i.e.* the model acts in accordance with the programmed logic behind it. In other

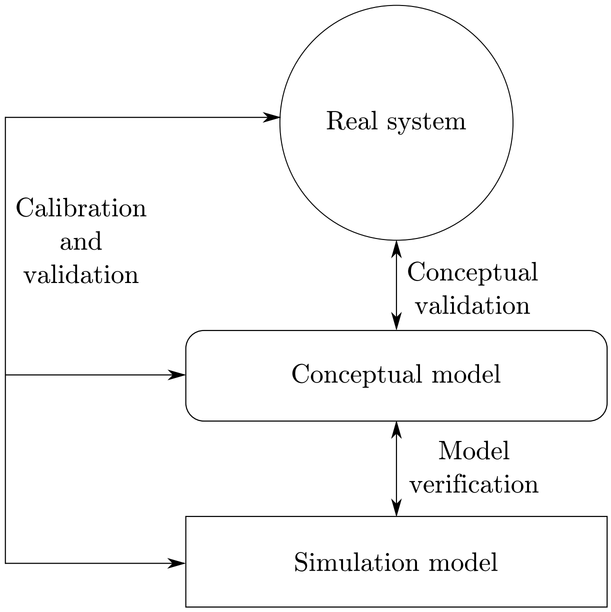


FIGURE 3.2: *The role of verification and validation in the simulation modelling process.*

words, verification involves testing whether the model implementation is free of any logical errors [40]. Balci [4] describes verification as "substantiating that the model is transformed from one form to another, as intended, with sufficient accuracy." The main method of verification therefore involves debugging the simulation model [68, 71], using a variety of possible techniques.

One of these techniques involves varying the input parameters over their allowable ranges and ascertaining whether the model produces acceptable output results [1, 53]. Making use of an interactive run control or debugger is also an effective way of potentially finding errors in a simulation model. This allows the modeller to trace variables throughout a simulation run, detecting when and under what conditions their values change, which may aid the modeller in discovering logical errors [1, 53]. Animation is also an effective verification tool which may be useful in ensuring that a model is working correctly, by providing a platform for observing entities that change states and travel through the system [1, 53]. This tool is particularly successful when combined with the tracing of variable values [1]. Another verification technique, sometimes referred to as a "sanity check," is an approach in which the results of the simulation model are analysed in order to determine whether the results are plausible and make logical sense [1].

The above-mentioned techniques are general procedures that may be followed when verifying a simulation model. Rakha *et al.* [62], however, state five verification steps that may be followed when dealing specifically with a traffic simulation model.

The first step is to select reasonable input parameters required for the model (such as a flow vehicle speed and a vehicle speed at capacity, for example).

The second step is an independent check to ensure that each value assigned to a model parameter agrees with typical real data, *i.e.* the value lies within an acceptable range. Suppose the typical free flow speed is between 80 km/h and 120 km/h, while the typical speed at capacity lies between 40 km/h and 90 km/h. In this case, an input value of 100 km/h as free flow speed and 50 km/h as speed at capacity does indeed lie within the typical real data range.

The third step is an additional test, ensuring that the combination of input parameters agrees with real data. Consider the case where the free flow speed is assigned a value of 80 km/h and the speed at capacity is 90 km/h. While both of these values agree with the check performed separately in the second step, this combination is not consistent with real data since the speed at capacity of a vehicle cannot be more than the free flow speed. This step is important as it ensures that various combinations of input parameters also agree with real data.

The fourth step involves the generation of results through the simulation model as well as through direct application of the logic without the use of the simulation model computer code.

The fifth and final step of the process is a comparison between the two outputs generated in the fourth step. If the difference in results are within an acceptable accuracy tolerance, the verification process is deemed successful. If not, changes should be made to the model and the process should be repeated.

3.5.2 Validation of a simulation model

The process of validating a model involves determining whether the design of a conceptual model accurately represents the real world system under consideration [6, 48]. According to Shannon [71], validation aims to answer three important questions: First, does the simulation model represent the actual system being studied, sufficiently? Secondly, does the model generate results that are consistent with real-world data? Finally, does the user have confidence in the results obtained from the model? Rakha *et al.* [62] state that validation attempts to confirm that the rules implemented in the model do indeed result in the predicted behaviour. Rakha *et al.* also describe validation in traffic simulation models specifically, as determining whether the rules followed by the model (*e.g.* rules for car following or lane changing) result in realistic quantities (*e.g.* capacities or queue lengths).

There are a number of methods for validating a simulation model. One of the most effective methods, however, involves comparing the output of the model to the output of a real system performing under the same conditions [48, 53]. This process is referred to as *results validation*, and is only possible when there is an actual real-world system against which to compare the model. The sets of output data may be compared using a variety of statistical tests in order to determine how significant the differences between the sets of output data are.

Another method of validation involves a thorough review of the results, where informed individuals evaluate the consistency of results and ensure that they are reasonable [48, 54, 64]. A sensitivity analysis may also be performed in which major factors that influence the performance of the model are identified [48, 64]. The use of Turing tests also offer validation, where an expert in the relevant field attempts to distinguish between the model output results and real system data [48, 64].

Validity should also be tested according to degenerate tests if possible [54, 64], where inappropriate input parameters are selected in order to test the degeneracy of the model. For example, if the arrival rate of customers at a single server is greater than the service rate, the size of the queue should increase over time. Another type of test in which input values are entered and the output evaluated is the *extreme condition test* [54, 64]. In this test, unlikely "extreme" values are entered into the system and the results are checked for reasonableness (for example, if the current resource inventory of a business is zero, then the production should also be zero).

Other methods of validation include making use of available historical data for comparison purposes, comparing the data obtained from the simulation model during independent runs and ensuring that there are no large discrepancies between runs, and making use of operational graphics for experts in the relevant field to observe and provide feedback on.

3.6 Traffic flow simulation

Traffic density cannot be described by certain departure times, routes and travel durations due to the fact that most of the traffic in a transportation network is due to unpredictable individuals, rather than scheduled transport vehicles and therefore these times, routes and durations cannot be known in advance [44]. As a result, accurate representations of traffic simulation are not easy to implement. According to Boxill and Yu [10], traffic simulation models are an important tool used to evaluate different scenarios, optimise traffic signals, predict future traffic conditions as well as aid traffic engineers in assessing and solving discovered problems.

Existing traffic simulation models may be classified in a number of ways, *i.e.* according to the level of abstraction or the nature of the road network (intersections, highways, *etc.*). In this section, a brief description is given of simulation models in the literature with different levels of abstraction. A discussion follows on the various advantages and disadvantages of using different traffic simulation software alternatives.

3.6.1 Types of traffic simulations

In this section, simulation models are classified according to their different levels of abstraction, namely *microscopic* traffic simulation models, *mesoscopic* traffic simulation models or *macroscopic* traffic simulation models. The difference between the level of abstraction in microscopic and macroscopic traffic simulation models is illustrated in Figure 3.3.

Microscopic traffic simulation

A *microscopic* traffic simulation model involves modelling vehicles individually, such that each vehicle is assigned unique movements (vehicle speed, acceleration and deceleration) and physical characteristics (vehicle length and position) [10, 38, 44]. The movement of a single vehicle through a road network is based on signal control, queue dissipation, a car following protocol, lane changing rules and gap acceptance algorithms [38]. An advantage of microscopic traffic simulation modelling is the high level of detail that can be incorporated into the model, allowing the operational level of the simulation to be observed [31].

Macroscopic traffic simulation

A *macroscopic* traffic simulation model, on the other hand, is less detailed and involves the analysis of aggregated vehicle data [10]. The movement of vehicles is typically modelled as a compressible fluid, making use of average vehicle flows, density and speed [36]. Due to the conservation of the number of vehicles, all macroscopic traffic models are based on the continuity of flow equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho V)}{\partial x} = V(x,t),$$

where $\rho(x,t)$ represents the density of vehicles at time t at position x along a roadway, while $V(x,t)\,\mathrm{d}x$ is the rate of vehicles joining or exiting the road network on road sections of length $\mathrm{d}x$ [28]. An advantage of macroscopic traffic simulation models over microscopic traffic simulation models is that the latter results in a lower computational cost due to a lower complexity of the model [36].

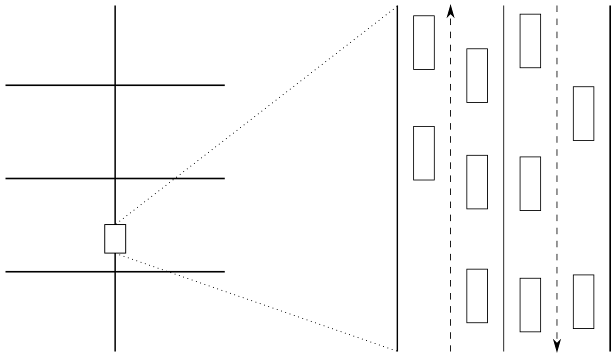


FIGURE 3.3: *A comparison between the level of detail in a macroscopic traffic model (left) and a microscopic traffic model (right).*

Mesoscopic traffic simulation

A *mesoscopic* traffic simulation model contains a combination of both microscopic and macroscopic elements in a traffic simulation model. The model retains the individual vehicle characteristics of the microscopic model, but a more aggregated approach is taken with respect to traffic dynamics [11]. In contrast to a microscopic traffic simulation model, the quantities associated with groups of objects (vehicles) are recorded, rather than those associated with individual entities. Unlike macroscopic traffic simulation models, mesoscopic traffic simulation models can accommodate any number of simultaneous groups of objects, as well as interactions between these objects [65]. Advantages of adopting a mesoscopic modelling approach, include a high degree of flexibility when it comes to the specification of input data and the fact that there are no complexity limitations when it comes to implementing algorithms in the mesoscopic model [65].

3.6.2 Simulation software packages

A number of traffic simulation software packages (microscopic, mesoscopic and macroscopic) are described briefly in this section. The advantages and disadvantages of these software packages are discussed as well as any specific functions they may have.

Microscopic traffic simulation software

Simulation of Urban Mobility (SUMO) [76] is an open-source microscopic simulation software package designed for large road networks. SUMO allows up to 10 000 roads to be incorporated in a road network, while maintaining a relatively low CPU and memory usage in comparison with other microscopic traffic simulation packages [43]. Due to the coarser nature of SUMO, it is not frequently used by transportation engineers to evaluate existing intersections. It is, however, well suited for evaluating the performance of traffic signal control algorithms due to its fast execution time [7]. Kotusevski and Hawick [43] found that SUMO is more writing intensive than other software, requiring the user to write multiple XML files without any kind of visualisation. This increases the chances of erring and thus Kotusevski and Hawick [43] expressed the opinion that SUMO is less user friendly and more difficult to use than other software. Another drawback of SUMO is that it does not yet support both left-hand and right-hand driving.

VisSim (a German acronym for "Traffic in cities — simulation model") is another example of a microscopic simulation tool, currently used by over 250 000 scientists and engineers worldwide

[73]. It is a behaviour-based multi-purpose software suite that is primarily used for optimising the flow of traffic, and is suitable for modelling both highway and urban traffic networks. VisSim includes a graphical user interface allowing users to add traffic data, signal data and construct road networks, while also including the animation of vehicular movements, traffic signals, detector actuation and a summary of the travel time of each vehicle [10]. In contrast to SUMO, VisSim is not open-source, although it does support both left-hand and right-hand driving.

CORSIM [55] is a microscopic traffic simulation software package which may be used for modelling traffic signal systems, freeway systems or a combination of both. It consists of an integrated set of two microscopic simulation models; one for modelling urban traffic and another for modelling traffic on highways and freeways. It allows users to define the traffic network, create traffic inputs and analyse different scenarios. Boxil and Yu [10] performed a study on an evaluation of traffic simulation model supporting intelligent transport systems and concluded that CORSIM had one of the highest probabilities of success in real-world applications.

Another microscopic traffic simulation software suite is Anylogic [3]. Anylogic is not designed specifically for traffic control, but it does include a road-traffic library (released in March 2016) that supports the agent-based modelling of vehicles in a traffic network. Users are able to drag road network elements (such as roads, traffic lights and stop lines) so as to easily build the network. Anylogic supports algorithmic implementation at signalised intersections, pedestrian movements, public transport and parking lots.

Macroscopic traffic simulation software

TRANSYT/10 (Traffic Network Study Tool) is an off-line macroscopic deterministic simulation platform that aims to optimise the offset, cycle length and phase split of traffic signals of a coordinated fixed-time control scheme in an attempt to reduce delays [63]. TRANSYT/10 is widely accepted as the standard method for setting signals in fixed-time control [10].

Another example of macroscopic simulation software is KRONOS, developed in the early 1980s. It is used mainly for freeway operation and is based on the *Lighthill-Whitham-Richards* (LWR) traffic flow theory [61]. The KRONOS software suite accounts for lane changing, acceleration, deceleration, merging, diverging and weaving, since these only occur at specific points along a freeway.

Mesoscopic traffic simulation software

One of the earliest mesoscopic simulation platforms is CONTRAM (continuous traffic assignment model) [72] and has been available commercially for over 20 years. It is used for traffic predictions — specifically traffic routes, queues and delays [66]. CONTRAM contains a number of tools that make it suitable for modelling diverse traffic situations and is specifically designed to model varying traffic demand [72].

INTEGRATION is a mesoscopic routing-orientated simulation that was also introduced early on. This simulation software represents traffic flow by means of individual vehicles, but each of these vehicles moves according to macroscopic flow theory, thus utilising both microscopic and macroscopic attributes. Car following and lane changing logic is incorporated as well as the modelling of toll plazas and vehicle emissions [80]. Real-time animation and vehicle statistics are also available for analysis purposes. Boxil and and Yu [10] found INTEGRATION to be the leading model for evaluating intelligent transport systems along corridors involving real-time traffic demand changes.

3.7 Chapter summary

This chapter opened in §3.1 with a number of definitions of concepts common to various simulation modelling approaches. This was followed by descriptions of the four prevailing types of simulation modelling paradigms in §3.2, namely agent-based modelling, discrete event modelling, system dynamics modelling and dynamical systems modelling. Both the advantages and disadvantages of simulation modelling were discussed in §3.3, while the twelve typical steps followed in a simulation study were given in §3.4. Simulation model verification and validation processes were described in §3.5 and the three levels of abstraction of traffic simulation models were finally discussed in §3.6. Associated software packages for each type of traffic simulation were also reviewed briefly.

CHAPTER 4

An agent-based traffic simulation model

Contents

The (novel) agent-based traffic simulation model designed and implemented for the purpose of this dissertation is described in detail in this chapter. The model was implemented in Anylogic 7.3.5 [3], making specific use of the Road Traffic Library that is included in the software. The chapter opens with a detailed description of the modelling framework in §4.1, with a focus on building the road network, implementing traffic signals, and generating vehicles as well as model output. This is followed by a verification and validation of the model in §4.2, while the components of an experimental design according to which the model is to be applied in the following chapters is described in §4.3. The chapter closes in §4.4 with a brief summary of the contents of the chapter.

4.1 Model framework

An agent-based, microscopic traffic simulation model was built for use as a test bed during an evaluation of traffic signal control algorithms in this dissertation. The model is capable of representing a real road network sufficiently accurately in order to compare the relative performances of the traffic signal control algorithms reviewed in Chapter 2 in a realistic fashion.

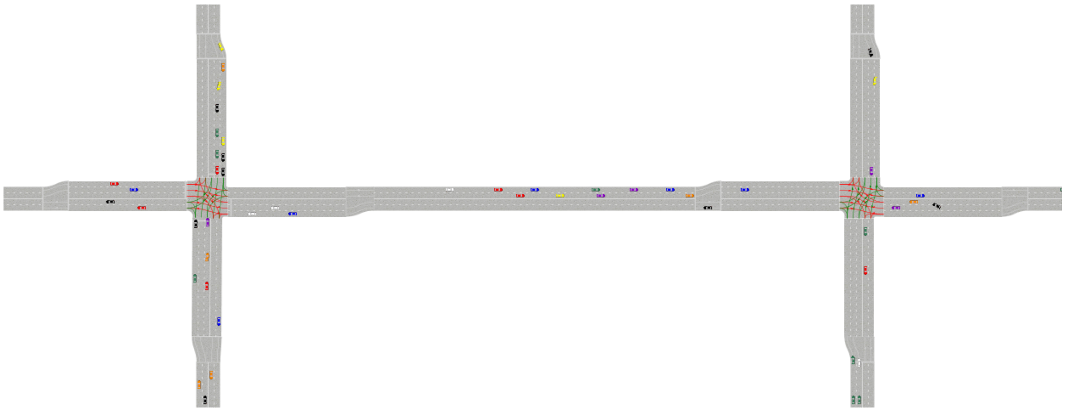


Figure 4.1: *A screen-shot from the simulation model of two neighbouring signalised intersections.*

While a number of changes are introduced to this model later in the dissertation, the original model is described here. A portion of a road network is depicted in Figure 4.1 as it is visualised in the simulation model. The model is stochastic in nature, as Monte Carlo methods as well as Poisson and uniform distributions are used to assign various attributes to vehicles. The model is also continuous, as well as dynamic, since the simulation model state evolves continually over time.

The static entities in the model are roads, intersections, stop lines and traffic signals, while vehicles are the only dynamic entities as they are the only entities that physically move as the simulation model runs over time. Traffic signals are implemented as a special type of entity as they also act as a resource, allocating green time to approaching vehicle flows. Each of these aforementioned entities has a number of attributes unique to it. Vehicle attributes are speed, acceleration, deceleration, colour, length, arrival rate, arrival location, destination, position, delay time, number of stops and distance travelled in the network, some of which are assigned to vehicles through the use of random distributions. The attributes of roads are length and number of lanes in each direction, while intersection attributes include the roads that are adjacent to them as well as which vehicle manoeuvres are permitted through the intersections. Current phase, remaining time of current phase, elapsed time of current phase, as well as phase durations and orders are the attributes of traffic signals, while the two attributes of a stop line are its position along a road and its associated road sign, if any.

Events can be either internal (endogenous) or external (exogenous). Exogenous events include the arrival of vehicles into the system, while endogenous events include vehicle manoeuvres, changes in vehicle speeds and the switching of traffic signals.

4.1.1 Constructing the road network

One of the first things to consider when building a road network within the model, is the desired scale. For a road traffic network specifically, a scale of 1 metre equivalent to four screen pixels is recommended by Anylogic [3]. This is also the scale adopted in this model. Road networks can be drawn free-hand by the user or else an image of a map (obtained from Google Earth [25], for example) may be imported into Anylogic after which a road network can be traced over it by the user. Road networks are constructed by dragging and connecting a number of space markup elements (such as roads, intersections, stop lines, bus stops and parking lots) using the Anylogic Road Traffic Library [3].

Roads may comprise straight or curved segments and contain a number of properties that have to be specified by the user, including whether they are one-way or two-way roads, the number of lanes in the specified "forward" direction as well as the number of lanes in the opposite or "backward" direction. A lane divider used to separate lanes in opposing directions by means of a barrier of a user-specified width, is also attributed to each road. Each road is "aware" of all vehicles that are located (or at least partially located) on it and it is possible to access these vehicles and their attributes. The road network as a whole also has a number of properties that are applied to all roads within it, including traffic flow direction, road appearance and lane width.

Intersections are used to connect two or more different roads, whether it be for traffic flows in multiple directions, or to gradually increase a two-lane road into a three-lane road. Movement through an intersection is controlled by lane connectors that define different paths through the intersection.

Stop lines are another method of controlling the behaviour of different traffic flow directions, requiring vehicles to stop there or pass through it. This entity also allows road sign behaviour (including an indication of the speed limit, the end of a speed limit or a yield) as well as facilitate the specification of code that will be executed each time a vehicle passes over the stop line.

The two distinct types of road networks considered in this dissertation include a 3×4 grid of signalised intersections, and a road corridor consisting of a number of signalised intersections, each connected to neighbouring intersections by means of straight road segments of equal length. Each intersection has a total of 20 lanes adjacent to it: twelve approach lanes (three approach lanes from four different directions) and eight exit lanes (two exit lanes to four different directions). All roads in the model are two-way, and the majority of the travel directions have two lanes. There is, however, a two-lane to three-lane expansion on approach lanes at 95 metres from each intersection. Properties of the road network include a left-hand driving direction as well as a specified lane width of 3.5 metres.

Intersections contain three approach lanes from each direction, with each of these lanes being associated with a different set of possible directions that vehicles travelling along them may move to, and these movements are defined by lane connectors, as may be seen in Figure 4.2. Possible movements along the leftmost lane include left turns and travelling straight through the intersection. The middle lane only allows movement of vehicles straight through the intersection, while the rightmost lane only permits right-turning at the intersection. The individual vehicle movement possibilities along these lanes are illustrated in Figure 4.3.

When a two-lane road expands into a three-lane road, an intersection (with only two directions) is required to connect them. These intersections are not controlled by signals and vehicles are not required to stop at the stop lines. Stop lines are employed, however, to keep track of vehicles that are in the intersection, and not on a road section. Intersections do not share the property of roads that allows an analyst to access vehicles in a road travelling in a specific direction. To alleviate this problem, stop lines are employed to execute simulation code each time a vehicle passes over the stop line, keeping track of the vehicle.

4.1.2 Traffic signals

In the Anylogic Road Traffic Library, traffic signals are incorporated which may be positioned at intersections, pedestrian crossings or anywhere that requires a method of controlling conflicting flows of traffic. This can be achieved through the use of intersection lane connectors or specified stop lines. Each traffic signal also contains a default ordered sequence of phases and phase

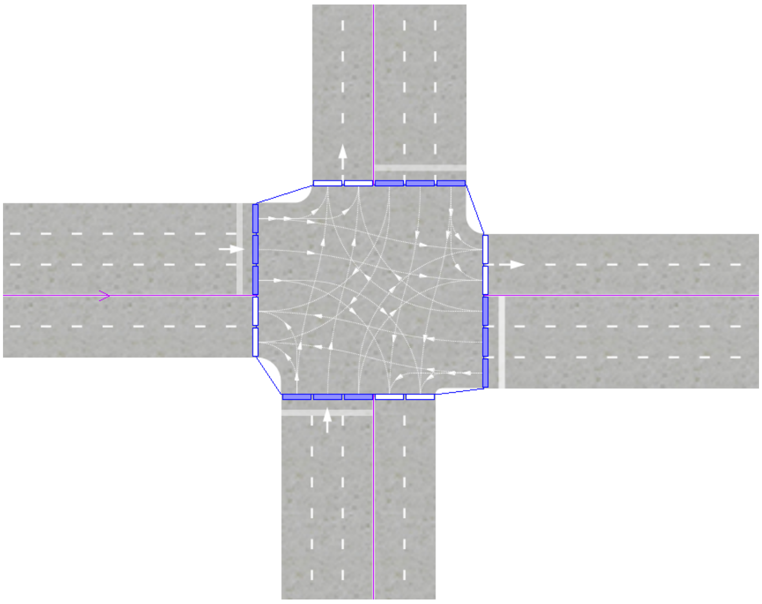
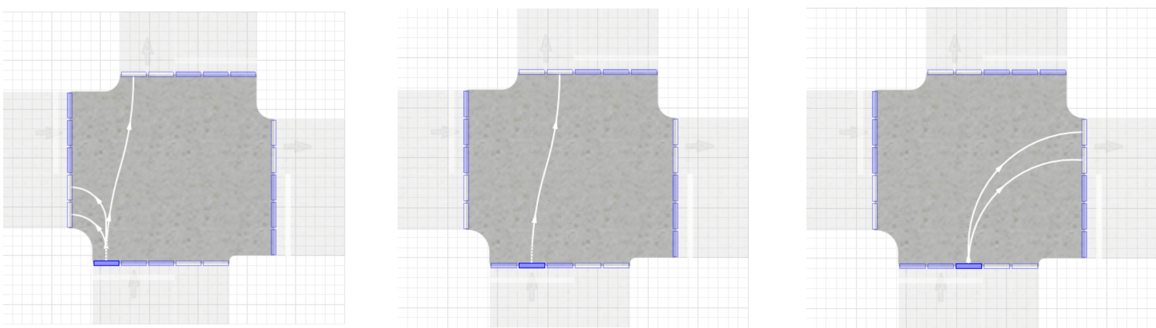


FIGURE 4.2: *A single intersection and its associated lane connectors, indicating traffic flow direction by means of arrow heads. Here the blue blocks at the end of each lane represent the approach lanes, while the white blocks represent the exit lanes.*



(a) (b) (c)

FIGURE 4.3: *The leftmost lane's travel directions are displayed in (a), while the path followed by vehicles emerging from the middle lane is displayed in (b). The possible lanes to exit from the rightmost lane are finally shown in (c).*

timings that have to be determined by the user. Once this sequence is determined, phases in the same sequence may be skipped if there is a lack of demand for them and thus the relative order of phases may change.

In the case of the two transport networks (the grid and corridor road networks) described above, each of the main intersections is modelled as self-interested agents which may or may not act independently of one another. These agents make decisions based on local information (*i.e.* the speed and distance of approaching traffic within the range of a SmartSensor [82]) in an attempt to achieve the natural emergence of global coordination across all intersections.

Six distinct phases make up the signal cycle of each intersection, as shown in Figure 4.4. These include a horizontal green phase, where all through and left-turning traffic in the horizontal direction (west to east and east to west) may travel through the intersection and right-turning vehicles in this direction may turn right on a permitted basis (*i.e.* they yield to traffic and only turn right if there are no approaching vehicles within a distance such that the vehicle may safely turn right), as illustrated in Figure 4.4 (a). The next green phase is a horizontal green right-turn phase, as shown in Figure 4.4 (b), which is an exclusive right-turn phase for vehicles turning right that are approaching from either an easterly or westerly direction.

Similarly, for the vertical direction (south to north and north to south), there is a vertical green phase that allows through and left-turning traffic to travel through the intersection, while right-turning vehicles may turn right on a permitted basis, as illustrated in Figure 4.4 (c). There is also a similar phase for right-turning vehicles in the vertical direction, where vehicles receive an exclusive green phase in the sense that they may turn without yielding, as shown in Figure 4.4 (d).

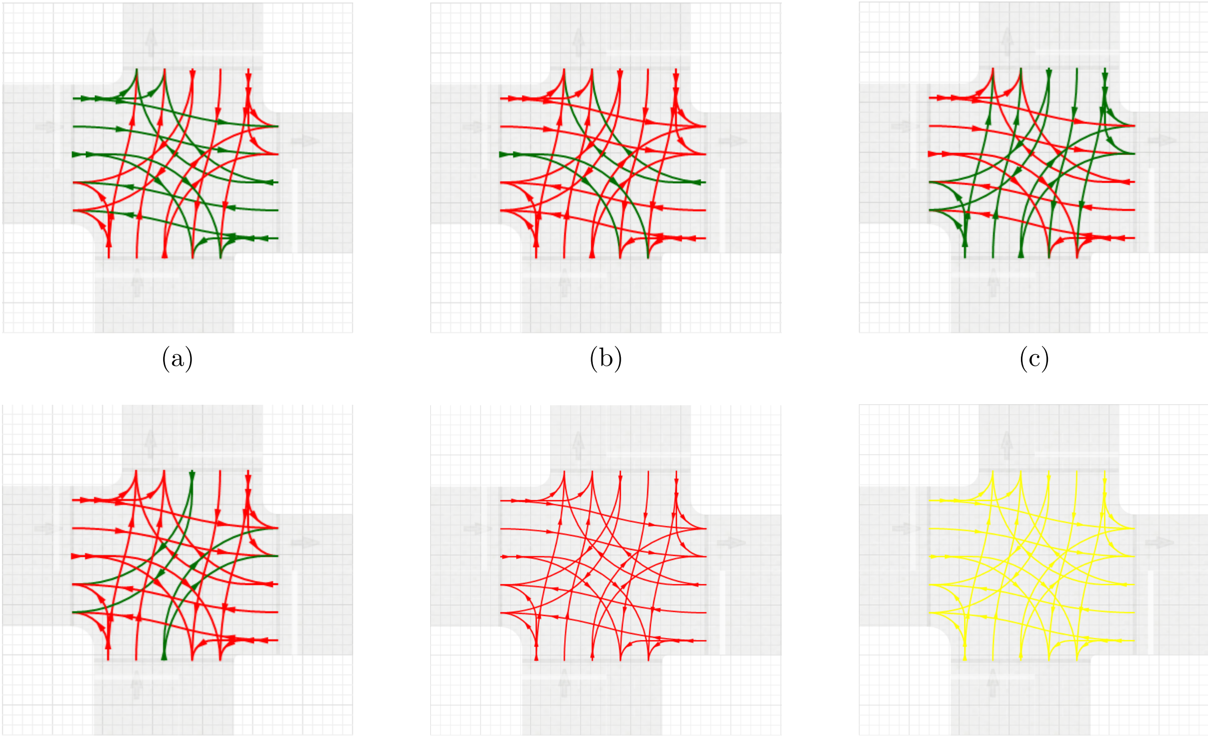
These four phases make up the ordered four distinct green phases of the signal cycle. There are also two additional phases that exist between these green phases. First there is an all-yellow phase (shown in Figure 4.4 (f)), which occurs immediately after the conclusion of a green phase, and acts as a 3-second warning to vehicles to begin slowing down as signals are about to change. The sixth and final phase is a 2-second all-red phase, shown in Figure 4.4 (e), allowing vehicles still in the intersection to pass safely through it before the next phase begins.

The duration and sequence of signal phases must be initialised by the user. These may, however, be programmed to change as the simulation model is executed. In the model, a number of different *events* exist, each representing one of the five selected algorithms reviewed in Chapter 2 which are to be compared in the following chapters. Events are used as a simple method of scheduling certain actions in the model that are only necessary for specific algorithms. These events may be triggered by a time-out, where actions are executed every so often or else they may be triggered once a certain condition is met.

The time-out triggered event is used for the O-TSCA algorithm, since demand, availability and throughput need to be updated regularly in order to implement this algorithm and thus are recalculated every 0.5 seconds. The condition-triggered event is employed in the implementation of the I-TSCA since green times are only recalculated after the previously allocated green time has elapsed. Once an event is triggered, a sequence of functions is executed in the model that determines values of importance, including green time durations, termination of green times as well as the skipping of phases.

4.1.3 Generating vehicles

Vehicles are generated and removed from a simulation run by means of a number of blocks linked by connectors in the Road Traffic Library. These include a *carSource* block where vehicle



(a) (b) (c) (d) (e) (f)

FIGURE 4.4: *The six different phases that make up the signal cycle (in no specific order).*

attributes are defined, a *carMoveTo* block which defines the destinations of vehicles and a *carDispose* block which removes vehicles from a simulation run.

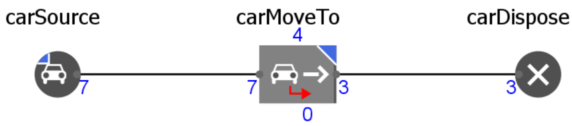


FIGURE 4.5: *A number of connected blocks in the simulation model, indicating that seven vehicles have been generated in total, four of which are still in the road network, while three have already reached their destinations and have therefore left the network.*

Vehicles enter the road network through any one of the twelve entering roads (three entering roads from the western direction, three entering roads from the eastern direction, four entering roads from the southern direction and four entering roads from the northern direction) along either a user-specified lane or a randomly allocated lane. It is possible to define vehicle arrivals in a number of different ways. They may be specified according to an arrival rate following a Poisson distribution with an input mean depending on the desired traffic volume or else the required interarrival times may be specified explicitly (these are the elapsed times between the arrivals of consecutive vehicles along a specific entering road). Vehicle arrivals may also be defined according to a deterministic arrival schedule in which case vehicles are generated at exact times according to a specified schedule, or else vehicles may enter the road network on specific calls of a vehicle inject function.

Once a vehicle has been generated, a number of unique attributes are simultaneously assigned to the vehicle. These include vehicle length, initial vehicle speed, preferred vehicle speed, maximum acceleration, maximum deceleration as well as the vehicle entry point and its destination in the simulation model run. Vehicle destinations and routes are defined according to a Monte Carlo simulation.

Vehicles obey all traffic signals. The model does not account for vehicles that run red signals or perform illegal manoeuvres. Vehicles keep a suitable following distance between one another that is stochastically calculated and based on the vehicles' decelerations. Stationary vehicle gaps are uniformly distributed distances ranging from 1 to 3 metres in length. If a vehicle approaches a conflicting flow of traffic, it automatically slows down and only travels through once it is possible to do so without colliding with another vehicle.

4.1.4 Model output data

Data recorded in the simulation model are saved and written to an excel file at the end of each simulation run. These data include six key *performance measure indicators* (PMIs) that are used to measure the relative performances of the traffic signal control algorithms embedded in the model.

The first of these PMIs is the mean delay time experienced by vehicles in the road network. This is calculated by subtracting the ideal time a vehicle spends in the system from the actual time spent by a vehicle in the system. The vehicle is aware of the distance it has travelled as well as its desired speed; the ideal travel time may therefore be calculated using these two values. The second PMI is the normalised mean delay time of vehicles, which is a ratio of the actual mean delay time experienced by vehicles to the ideal travel time of a vehicle. If a vehicle has a normalised mean delay of 2.0, it means that a vehicle took twice as long to reach its destination than it would have had it travelled at its desired speed the entire way.

The third PMI is the average number of stops made by a vehicle, and is calculated by incrementing a counter each time a vehicle comes to a complete stop. The fourth PMI is the normalised average number of stops a vehicle makes. This takes into account the number of intersections that a vehicle passes through, since a vehicle that travels through four intersections is likely to stop more often than a vehicle that only travels through one intersection.

The previous two PMIs are only indications of when a vehicle comes to a complete stop, and do not take into account vehicles that travel very slowly. In order to account for this deficiency, the mean time that a vehicle spends travelling under 10km/h is also recorded. A sixth PMI is introduced and is referred to the normalised time a vehicle spends under 10km/h, in order to represent the percentage of time a vehicle spends travelling at a very slow pace. This value is calculated by dividing the time a vehicle spends travelling under 10km/h by the total length of time it spends in the network.

Additional output produced by the simulation model includes the horizontal and vertical average green times executed at each intersection. In the case where a maximum green time is included in the algorithmic implementation, the number of times this maximum value was reached and caused a signal change, is additionally recorded. The average saturation of the road network is also recorded, which is the ratio of the effective space occupied by vehicles to the total amount of space that is available in the road network.

For all the aforementioned output generated by the simulation model, further information is also recorded in addition to the mean values. This includes the corresponding minimum values, maximum values, standard deviations and confidence intervals, as well as the number of data points included in these calculations.

Anylogic [3] allows the user to observe visualisations of the simulation model runs. Variables that change over time, as well as a variety of other information pertaining to a specific vehicle, can be accessed and tracked while the simulation model is executed. This allows the analyst to perform a real-time analysis if necessary.

4.2 Model verification and validation

In this section, it is described how some of the verification and validation techniques mentioned in §3.5 were applied to the simulation model described in §4.1. These techniques were not performed only once, but iteratively, and throughout the entire process of the building of the model.

4.2.1 Verification of the traffic simulation model

Verification is important as it ensures that a model performs as is expected and is free of logical errors. While there are many possible techniques by which to verify a simulation model, the major methods that were employed in this study are described in this section.

Anylogic [3] contains both an *interactive run controller* (IRC) and a debugger. When model source code is compiled, the debugger searches through the code and reports any errors that it detects. If an error is found, the model is not executed and a description of the error is given. The location of the error in the code is specified and possible explanations are provided by the debugger which may account for the cause of the error. If the code is compiled successfully, the model may be executed. There are two types of runtime errors that may occur during a simulation run: Java exceptions or simulation errors. Java exceptions may occur when Java code written by the user contains computational errors (such as dividing by zero or accessing a null pointer), while simulation errors are caused by programmed errors in logic. An example of a simulation error is when a vehicle is generated along a road that contains no permitted route to its assigned destination. If this specific error occurs, the simulation run is aborted and a red circle is displayed at the entry point of the relevant vehicle, as shown in a screen-shot of the simulation model in Figure 4.6.

Another prominent aspect of the verification process is the animation tool that exists in the Anylogic [3] software suite. Animation allows the user to detect any unexpected characteristics exhibited by the model (such as unexpected vehicle behaviour, appearance or interaction with the road network).

Variable tracing and print statements also played a major role in verifying that the self-organising algorithms reviewed in Chapter 2 were implemented correctly. Variables may be set to “visible” in the simulation, so that their values may be monitored visually throughout a simulation run. This was particularly useful for the implementation of the O-TSCA (illustrated in Figure 4.7). In the O-TSCA, signals change once the throughput has exceeded the demand and the pressure of the direction receiving service is less than the pressure of the competing direction. The variable may be tracked during a simulation run so as to ensure that the signals do indeed change at the correct point in time.

Similarly, print statements were central to ensuring that the I-TSCA performed as intended. In the I-TSCA, each vehicle’s position, current speed, desired speed, predicted time to reach

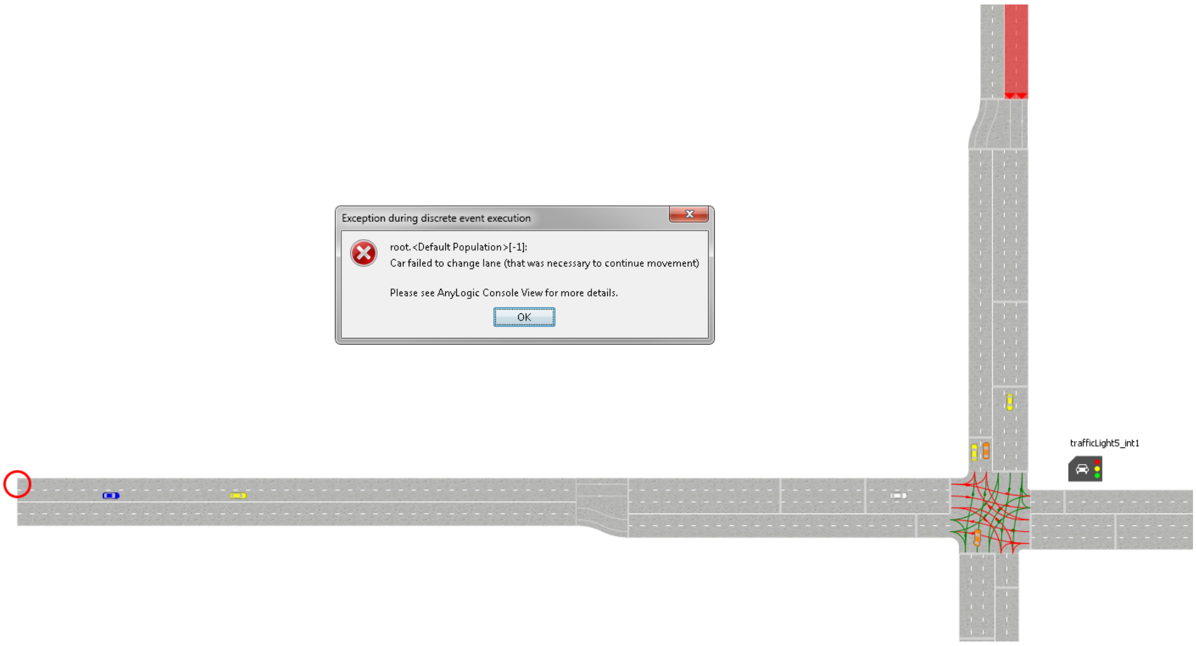


FIGURE 4.6: *A vehicle is generated and expected to enter the network at the red circle. The exit road is, however, indicated by the red road segment and it is apparent that there is no route for the vehicle to follow in order to reach this lane, as it contains traffic flowing from an opposing direction. This example represents the user specifying an incorrect exit road direction (it is more likely that the user intended the vehicle to exit on the left side of the road).*

the intersection and predicted delay time were printed out. This made it possible to ascertain whether the predicted delay was calculated correctly. Each time the I-TSCA re-evaluated the delay time "costs," the predicted delays were all printed out and the total cost for switching signals and extending a green signal were printed, as were the individual vehicle delays, in order to ensure that the total cost calculation was correct.

4.2.2 Validation of the traffic simulation model

Successful validation of a model results in a system that accurately depicts the real system being modelled. Three different techniques described in §3.5.2 were employed in this study to ensure a valid simulation model.

The first of these methods was a sensitivity analysis, altering central parameters that influence the performance of the model, and ensuring that the resulting behaviour of the model was as expected. An example of this involved modifying the arrival rate and observing the change it had on the vehicle occupation of the road network. If all other parameters are kept constant (*e.g.* traffic signal timings remain unchanged) and the arrival rate is increased, it is anticipated that vehicle queues will grow, that the saturation level of the network will increase and that the average vehicle delay will increase. If, on the other hand, arrival rates are lowered, the opposite is expected to occur. The occurrence of these expectations were verified in a number of simulation runs.

The second validation method involved the analysis and interpretation of the simulation results. The results of the simulation model output contain randomness due to the stochastic nature of the model arrival rates, vehicle speeds and destinations. The model is therefore expected to yield different output data during each simulation run. These data should, however, not contain

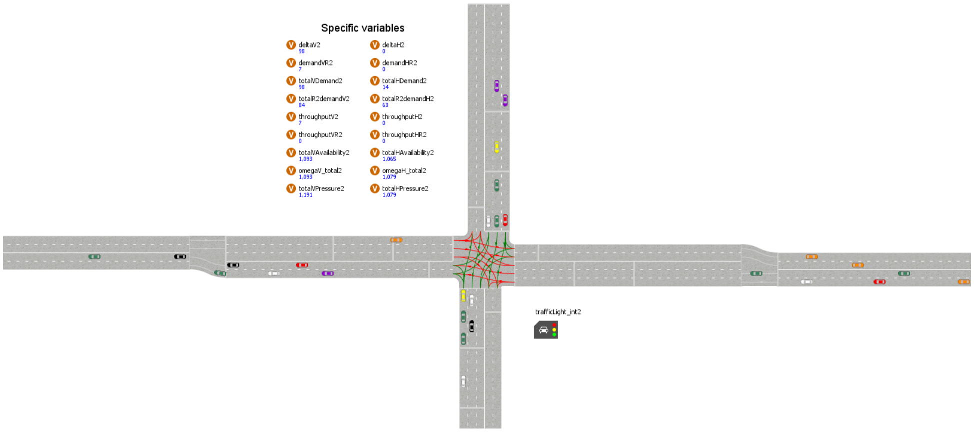


FIGURE 4.7: *The relevant variables necessary for controlling the switching logic of the O-TSCA are displayed at each intersection for verification purposes.*

undue variance. The variance of the results were analysed to ensure that there were not large discrepancies between the output data of simulation runs. The results were also analysed to ensure that the values made physical sense. It was, for example, verified that vehicle delay times were not unlikely values (*e.g.* negative or very large values).

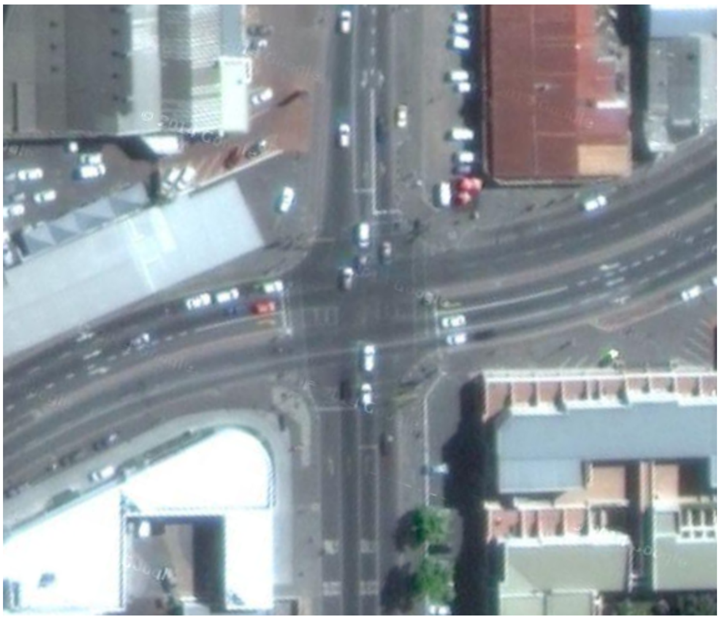


FIGURE 4.8: *An isolated intersection in Stellenbosch where Adam Tas and Bird Streets intersect. The approach on the left hand side is referred to as Adam Tas (AT), the approach on the right is referred to as R44, the bottom approach is referred to as Bird and the approach from the top is referred to as N1.*

The third and final method of validation (and possibly the most important) entailed comparing the simulated output to the output of a real system. The real data used were collected for a previous study by Van der Merwe [81] at an isolated traffic intersection in Stellenbosch, shown in Figure 4.8. The intersection has four approaches, consisting of twelve approach lanes and six exit lanes. The legal manoeuvres for each lane are as follows: left lanes permit both turning and travelling straight for the horizontal direction (AT and R44), yet only left-turning is permitted for the left lane in the vertical directions (Bird and N1). The centre lane only allows travelling straight for all approaches, while the right lane only permits right-turning for all approaches.

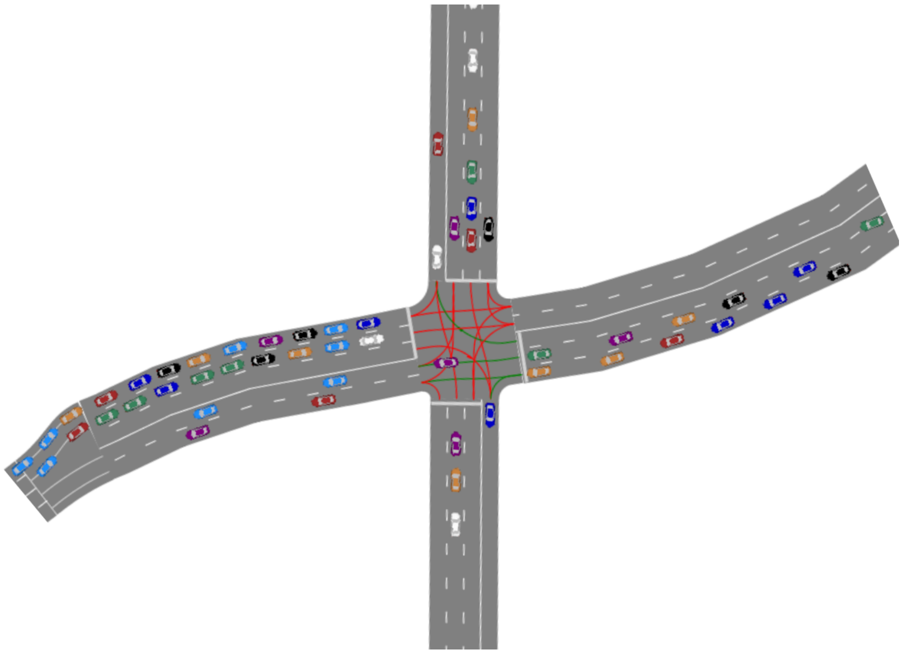


FIGURE 4.9: *A screen-shot taken of the simulation model of §4.1 for the intersection in Figure 4.8 during a simulation run in the Anylogic software suite.*

The intersection is modelled in Anylogic [3], and a screen-shot is shown in Figure 4.9.

The first phase allocates all vehicles travelling from the R44 direction a green signal, and the second phase assigns all the vehicles travelling along the horizontal direction a green signal (R44 and AT), while right-turning occurs on a permissive basis. The third phase provides the entire vertical direction with green time, while right-turning vehicles do so on a permissive basis once again. The fourth and final phase allows protected right-turn phases for vehicles travelling from Bird and N1, while left-turning vehicles from AT and R44 are also permitted to complete their turns.

The numbers of vehicles passing through the intersection, as well as their associated manoeuvres, were recorded by Van der Merwe [81] in fifteen minute intervals from 06:30 until 18:00 on a Tuesday during school and university term time in order to capture the standard traffic conditions at the signalised intersection. For the purpose of the validation of the model described in §4.1, the vehicles passing through the intersection were aggregated into eleven one-hour periods and one half-hour period (as were their associated manoeuvre probabilities). Similarly, the lengths of the green time phases recorded by Van der Merwe [81] were aggregated into morning, midday and afternoon green times and are summarised in Table 4.1.

The average hourly arrival rates, as well as turning probabilities for each approach, were taken as input for the simulation model. The simulation was executed for eleven and a half simulation hours, recording output data for each hour, and was replicated 30 times. The average output results of the 30 replications were compared against the actual known values and the absolute errors were recorded, as shown in Tables 4.2–4.13. The largest discrepancy found between the actual data and the model output was for the left-turning vehicles from the R44 approach during the second hour, achieving a 7.3% error. The total simulated number of vehicles passing through the intersection, however, was found to be only 2% less than the actual value, indicating that the model described in §4.1 accurately depicts the real-world system to which it was compared.

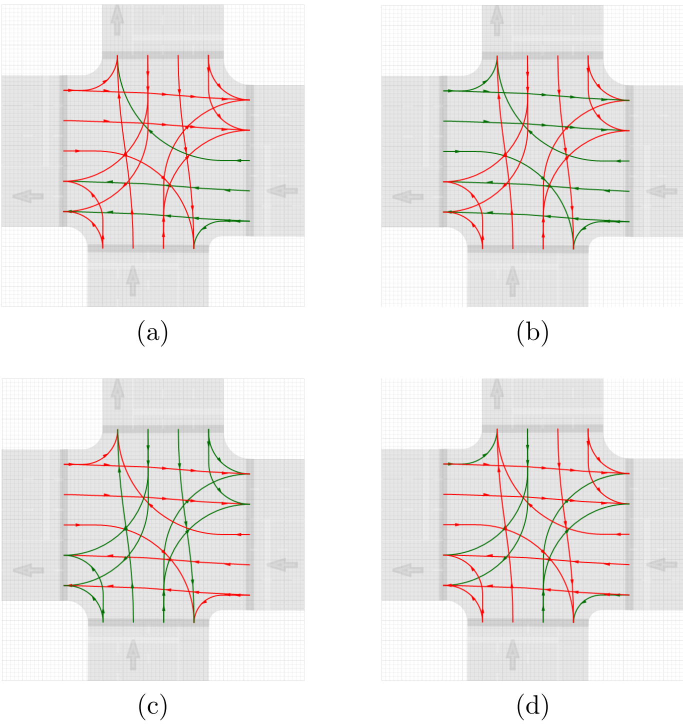


FIGURE 4.10: *The four green phases that make up the signal cycle of the intersection in Figure 4.8. The first phase is represented by the signals in (a), the second phase is represented by the signals in (b), the third phase is represented by the signals in (c) and the fourth phase is represented by the signals in (d).*

Period	First phase	Second phase	Third phase	Fourth phase
Morning (06:30–09:30)	10.00	55.14	55.79	14.71
Midday (09:30–16:30)	12.38	46.75	68.88	9.75
Afternoon (16:30–18:00)	8.11	52.84	70.74	3.79

TABLE 4.1: *Length of four different green phases and their associated green times throughout the course of the day (measured in seconds).*

Approach	Actual	Simulation after 1 hour	% error
AT(L)	255	238.8	6.4
AT(S)	503	498.6	0.9
AT(R)	7	7.4	5.3
R44(L)	195	184	5.7
R44(S)	625	614.6	1.7
R44(R)	128	120.4	5.9
Bird(L)	5	5.3	5.3
Bird(S)	216	219.5	1.6
Bird(R)	99	99.3	0.3
N1(L)	31	30.8	0.7
N1(S)	447	449.4	0.5
N1(R)	347	329.9	4.9
Total	2 858	2 798	2.1

TABLE 4.2: *After 1 simulation hour.*

Approach	Actual	Simulation after 2 hours	% error
AT(L)	449	426.1	5.1
AT(S)	986	983.9	0.2
AT(R)	20	20.2	1.2
R44(L)	555	514.4	7.3
R44(S)	1 286	1 268.9	1.3
R44(R)	279	262.3	6.0
Bird(L)	16	15.7	1.6
Bird(S)	539	537.1	0.3
Bird(R)	302	295.4	2.2
N1(L)	77	74.5	3.3
N1(S)	982	972	1.0
N1(R)	667	628.7	5.7
Total	6 158	5 999.3	2.6

TABLE 4.3: *After 2 simulation hours.*

Approach	Actual	Simulation after 3 hours	% error
AT(L)	672	641.5	4.5
AT(S)	1 363	1 358.3	0.3
AT(R)	37	37	0.1
R44(L)	762	714.4	6.2
R44(S)	1 643	1 624.4	1.1
R44(R)	367	348.2	5.1
Bird(L)	36	35.9	0.3
Bird(S)	825	827.4	0.3
Bird(R)	465	455.3	2.1
N1(L)	133	129.5	2.6
N1(S)	1 436	1 428.8	0.5
N1(R)	908	860.2	5.3
Total	8 647	8 460.9	2.2

TABLE 4.4: *After 3 simulation hours.*

Approach	Actual	Simulation after 4 hours	% error
AT(L)	872	834.9	4.3
AT(S)	1 742	1 734.4	0.4
AT(R)	56	55.1	1.6
R44(L)	963	909.1	5.6
R44(S)	1 963	1 941.2	1.1
R44(R)	446	425.6	4.6
Bird(L)	46	45.9	0.2
Bird(S)	1 136	1 137.9	0.2
Bird(R)	626	615.2	1.7
N1(L)	187	182.7	2.3
N1(S)	1 829	1 819.5	0.5
N1(R)	1 148	1 092.9	4.8
Total	11 014	10 794.4	2.0

TABLE 4.5: *After 4 simulation hours.*

Approach	Actual	Simulation after 5 hours	% error
AT(L)	1 086	1 040.8	4.2
AT(S)	2 152	2 141.7	0.5
AT(R)	68	67.7	0.5
R44(L)	1 153	1 094.6	5.1
R44(S)	2 285	2 261	1.1
R44(R)	532	510	4.1
Bird(L)	61	60.9	0.2
Bird(S)	1 440	1 441.4	0.1
Bird(R)	804	789.8	1.8
N1(L)	249	243.6	2.2
N1(S)	2 184	2 173.6	0.5
N1(R)	1 366	1 304.7	4.5
Total	13 380	2 798	1.9

TABLE 4.6: *After 5 simulation hours.*

Approach	Actual	Simulation after 6 hours	% error
AT(L)	1 298	1 245.8	4.0
AT(S)	2 524	2 511.8	0.5
AT(R)	90	88.9	1.3
R44(L)	1 351	1 286.4	4.8
R44(S)	2 616	2 589.9	1.0
R44(R)	620	595.8	3.9
Bird(L)	75	74.9	0.1
Bird(S)	1 767	1 768.4	0.1
Bird(R)	985	968.3	1.7
N1(L)	327	312.6	4.4
N1(S)	2 514	2 534.0	0.8
N1(R)	1 625	1 531.4	5.8
Total	15 792	15 508	1.8

TABLE 4.7: *After 6 simulation hours.*

Approach	Actual	Simulation after 7 hours	% error
AT(L)	1 531	1 470	4.0
AT(S)	2 938	2 923.3	0.5
AT(R)	111	109.1	1.7
R44(L)	1 553	1 482.3	4.5
R44(S)	3 008	2 979.8	0.9
R44(R)	703	677	3.7
Bird(L)	101	100.4	0.6
Bird(S)	2 140	2 139.8	0.0
Bird(R)	1 199	1 177.6	1.8
N1(L)	387	370.5	4.3
N1(S)	2 919	2 933.8	0.5
N1(R)	1 893	1 788.7	5.5
Total	1 8483	18 152.2	1.8

Table 4.8: *After 7 simulation hours.*

Approach	Actual	Simulation after 8 hours	% error
AT(L)	1 751	1 681.8	4.0
AT(S)	3 341	3 322.8	0.5
AT(R)	133	130.3	2.0
R44(L)	1 742	1 665.3	4.4
R44(S)	3 378	3 348.6	0.9
R44(R)	804	774.8	3.6
Bird(L)	120	119.1	0.7
Bird(S)	2 508	2 507.4	0.0
Bird(R)	1 410	1 382.5	1.9
N1(L)	412	395.4	4.0
N1(S)	3 176	3 191	0.5
N1(R)	2 080	1 972.8	5.2
Total	20 855	20 491.9	1.7

Table 4.9: *After 8 simulation hours.*

Approach	Actual	Simulation after 9 hours	% error
AT(L)	1 995	1 913.8	4.1
AT(S)	3 805	3 783	0.6
AT(R)	146	143	2.0
R44(L)	1 940	1 855.8	4.3
R44(S)	3 803	3 769.3	0.9
R44(R)	901	868.8	3.6
Bird(L)	139	137.6	1.0
Bird(S)	2 918	2 916.4	0.1
Bird(R)	1 629	1 594.2	2.0
N1(L)	464	446.2	3.8
N1(S)	3 542	3 554.7	0.4
N1(R)	2 296	2 182.8	4.9
Total	23 578	23 165.7	1.7

Table 4.10: *After 9 simulation hours.*

Approach	Actual	Simulation after 10 hours	% error
AT(L)	2 232	2 142.9	4.0
AT(S)	4 183	4 160.9	0.5
AT(R)	170	166	2.3
R44(L)	2 136	2 045.1	4.3
R44(S)	4 224	4 188	0.9
R44(R)	1 021	985.5	3.5
Bird(L)	159	157.1	1.2
Bird(S)	3 383	3 380.1	0.1
Bird(R)	1 836	1 794.8	2.2
N1(L)	506	487.4	3.7
N1(S)	3 901	3 913.6	0.3
N1(R)	2 520	2 399.2	4.8
Total	26 271	25 820.5	1.7

Table 4.11: *After 10 simulation hours.*

Approach	Actual	Simulation after 11 hours	% error
AT(L)	2 497	2 395	4.1
AT(S)	4 709	4 685.7	0.5
AT(R)	190	185.2	2.5
R44(L)	2 415	2 314.1	4.2
R44(S)	4 611	4 576.1	0.8
R44(R)	1 130	1 091.7	3.4
Bird(L)	169	166.3	1.6
Bird(S)	3 966	3 935.4	0.8
Bird(R)	2 228	2 121	4.8
N1(L)	560	540	3.6
N1(S)	4 442	4 452.8	0.2
N1(R)	2 724	2 591.1	4.9
Total	29 641	29 054.2	2.0

Table 4.12: *After 11 simulation hours.*

Approach	Actual	Simulation after 11.5 hours	% error
AT(L)	2 659	2 533.1	4.7
AT(S)	5 028	4 980.2	1.0
AT(R)	194	188.6	2.8
R44(L)	2 559	2 450.3	4.2
R44(S)	4 815	4 777.9	0.8
R44(R)	1 176	1 137.3	3.3
Bird(L)	173	169.9	1.8
Bird(S)	4 202	4 159.5	1.0
Bird(R)	2 415	2 281.5	5.5
N1(L)	593	571.6	3.6
N1(S)	4 736	4742	0.1
N1(R)	2 820	2 684.6	4.8
Total	31 370	30 676.4	2.2

Table 4.13: *After 11.5 simulation hours.*

4.3 Experimental design

In this section, various aspects of the experimental design according to which the five algorithms reviewed in Chapter 2 are to be compared later in this dissertation, are discussed. This includes discussions on the calculation of a suitable simulation warm-up time and a number of general specifications of the road network, such as specific vehicle and road attributes. The types of statistical analysis performed in respect of the simulation model output data collected from various simulation runs are also described in this section.

4.3.1 The simulation warm-up period

At commencement of the simulation model execution, there are initially no vehicles in the road network, and over a period of time the number of vehicles in the network gradually increases until a stable average number of vehicles in the network is reached. It would be inaccurate to record various output data before the number of vehicles in the network have stabilised, as this would imply that traffic conditions are initially lighter than they actually were, resulting in misleading or inaccurate model output. For this reason it is necessary to determine a suitable simulation warm-up time, long enough to ensure consistency, yet short enough to avoid wasted time during simulation runs.

The method described by Law [50] is employed in this dissertation to determine an appropriate length of the warm-up period.

Let $Y_1, Y_2, Y_3, \ldots$ be observations of the number of vehicles Y in a simulation model run at discrete points $1, 2, 3, \ldots$ in time, respectively. The steady state mean of Y is given by

$$\bar{m} = \lim_{t \to \infty} E(Y_i). \tag{4.1}$$

Law [50] states, however, that an initial finite time period x exists during which (4.1) does not hold (*i.e.* when $E[\bar{Y}(x)] \neq \bar{m}$). It is suggested that a warm-up period $[1, x^*]$ be introduced, and that all observations made during this period are to be disregarded. A better estimation of $\bar{m}$ is thus

$$\bar{Y}(x, x^*) = \frac{\sum_{i=x^*+1}^{x} Y_i}{x - x^*} \tag{4.2}$$

as opposed to $\bar{Y}(x) = \frac{\sum_{i=1}^{x} Y_i}{x}$. The following four steps are suggested by Law [50] in order to determine a suitable warm-up period $[1, x^*]$:

1. Run the simulation model ω times, each for a length of x time units. The resulting output Y_{ij} represents the i^{th} observation from the j^{th} model run, for $i = 1, 2, \ldots, x$ and $j = 1, 2, \ldots, \omega$.

2. Calculate the average of the observations by dividing their sum by the number of simulation runs, *i.e.* $\bar{Y}_i = \sum_{j=1}^{\omega} Y_{ij}/\omega$ for $i = 1, 2, \ldots, x$. The average value $\bar{Y}_i$ has mean $E(\bar{Y}_i) = E(Y_i)$ and variance $\text{Var}(\bar{Y}_i) = \text{Var}(Y_i)/\omega$ for all $i = 1, 2, \ldots, x$.

3. Calculate a moving average using a window over the averaged processes $\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_x$ in order to remove high frequency oscillations. The moving average is given by

$$\bar{Y}_i(y) = \begin{cases} \frac{\sum_{s=-y}^{y} \bar{Y}_{i+s}}{2y+1}, & \text{if } i = y+1, \ldots, x-y \\ \frac{\sum_{s=-(i-1)}^{i-1} \bar{Y}_{i+s}}{2i-1}, & \text{if } i = 1, \ldots, y, \end{cases} \tag{4.3}$$

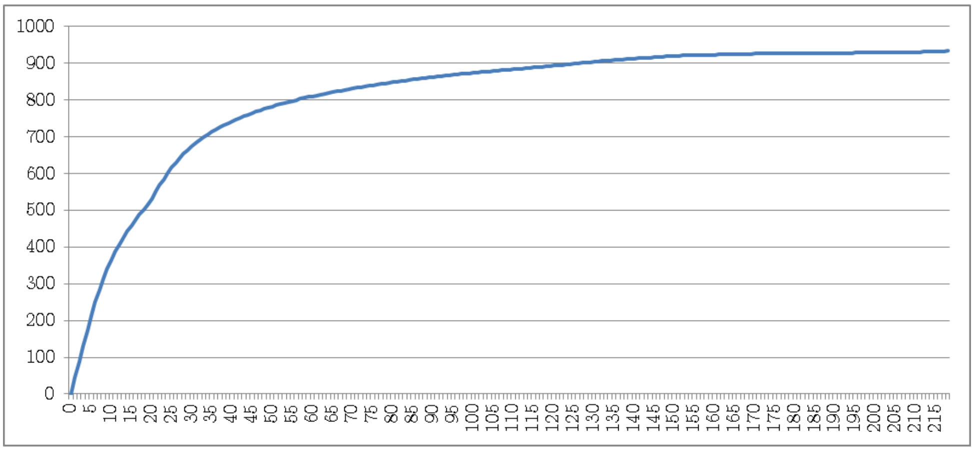


FIGURE 4.11: *An indication of the simulation warm-up time under heavier traffic conditions (an arrival rate of 20 vehicles per minute). The warm-up time is approximately 1 800 seconds, while the steady state number of vehicles in the system is approximately 933 vehicles.*

where y is the size of the moving window average and is selected such that $0 < y \leq x/4$.

4. The warm-up period x^* is chosen as the value of i for which mean values $\bar{Y}_i(y), \bar{Y}_{i+1}(y), \ldots, \bar{Y}_{x-y}(y)$ have converged to a constant value.

In the simulation model of §4.1, the value of ω was chosen to be 30 replications as this value gave an accurate indication of the steady state of the system. Each simulation was run for 2 200 seconds and observations were made every 10 seconds (hence there were 220 observations). It was found that for both lighter and heavier traffic conditions, a warm-up period of 1 800 seconds was sufficient, as was found by Einhorn [16]. A graph depicting the convergence of the number of vehicles in the road network over time is shown in Figure 4.11.

4.3.2 General specifications of simulation framework

In the simulation model of §4.1, vehicle arrivals are determined according to an arrival rate following a Poisson distribution with an input mean depending on the desired traffic volume, which is equivalent to an exponentially distributed interarrival time with mean equal to the reciprocal of the arrival rate. A mean of $\lambda = 10$ vehicles per minute per entry point to the road network represents a lighter traffic demand, whereas a mean of $\lambda = 20$ vehicles per minute represents a heavier traffic demand in this dissertation. These values were settled upon according to the arrival rates observed in the validation study in §4.2.2 using the data collected by Van der Merwe [81]. The lighter traffic flow rate was obtained by averaging the arrival rates of each approach of the hour during which the lowest total traffic density was observed (10:30–11:30). The averaging of these values yielded an hourly arrival rate of 591.5 vehicles which corresponds to an arrival rate of $9.9 \approx 10$ vehicles per minute, as shown in Table 4.14. Initially, heavier traffic conditions were taken as the largest average number of vehicles that passed through the intersection per hour, which occurred during the period 18:00–18:30. These averages resulted in an arrival rate of 14.4 vehicles per minute, which when implemented in the simulation model, did not appear to yield substantially heavier traffic conditions than the lighter traffic demand. Thus, the largest observed hourly rate of 1 172 vehicles per hour (equivalent to an arrival rate of $19.5 \approx 20$ vehicles per minute) is used to represent heavier traffic conditions. Since there are 14 points of entry to the grid network, a lighter traffic demand corresponds to 140 vehicles entering

	Lowest arrival rate		Highest arrival rate	
Approach	Hourly arrival rate	Arrival rate per minute	Hourly arrival rate	Arrival rate per minute
AT	636	10.6	970	16.2
R44	598	10.0	788	13.1
Bird	497	8.3	854	14.2
N1	635	10.58	846	14.1
Average	591.5	9.9	864.5	14.4

TABLE 4.14: *The arrival rates of vehicles at the intersection in Figure 4.8. The smallest hourly average arrival rate occurs during the hour 10:30–11:30, while the largest hourly arrival rate occurs during the period 18:00–18:30.*

the network per minute, on average, while a heavier demand corresponds to 280 vehicles entering the grid per minute, on average.

The physical length of vehicles is assigned a value of 5 metres, which is a realistic length of an actual vehicle. While it would be more realistic to include larger vehicles, such as buses and large trucks, this deficiency is not expected to affect the relative performance of the algorithms in any way. It is suggested as future work to include varying vehicle lengths as well as associate slower speeds with larger vehicles. The minimum safety gap between vehicles is assumed to be 2 metres, as the Anylogic software suite assigns a uniform random number between 1 and 3 metres between stationary vehicles. The effective length of each vehicle in the simulation model is the sum of the actual vehicle length and its safety gap, which is 7 metres.

The speed limit in the simulation model is taken as 60 km/h and individual vehicle speeds are assigned randomly according to the uniformly distributed probability density function 60 $\times$ uniform(0.7,1.2), allowing a range of speeds from 42 km/h to 72 km/h.

In the grid road network topology, each vehicle has either seven or nine possible locations to exit the road network, depending on whether the vehicle is generated on a horizontal or vertical entry road. If the vehicle is generated to enter from an easterly or a westerly direction, possible exits include the exit road straight ahead, or any one of the eight roads ahead if the vehicle were to make a right or left turn, forming a total of nine possible exit routes. Similarly, for vehicles generated facing south or north, possible exits include the road straight ahead, or else one of the six roads that would lie ahead if the vehicle were to make a right or left turn, totalling seven possible exits. These exit directions are illustrated in Figure 4.12.

Similarly, in the corridor topology each vehicle has either $2n+1$ or 3 possible routes through the network (where n represents the number of intersections in the corridor), depending on whether the vehicle enters along the corridor or from a side road, respectively. If a vehicle enters along the corridor, possible exit roads include the end of the road straight ahead or any one of the left or right turns the vehicle could make. On the other hand, if a vehicle enters the network from a side road, the vehicle can either turn left, right or continue straight, resulting in three possible destinations. These vehicle generation and destination pairs are illustrated in Figure 4.13. In both the grid and corridor road network, a vehicle is assumed to turn at most once (assuming that no roads are closed).

Only horizontal and vertical phases are competing for green time in the experiments in which the traffic signal control algorithms are compared later in this dissertation. The protected right-turn phases are only employed when four or more vehicles remain after receiving green time, where these specific vehicles were allowed to turn on a permitted basis. The protected right-turn

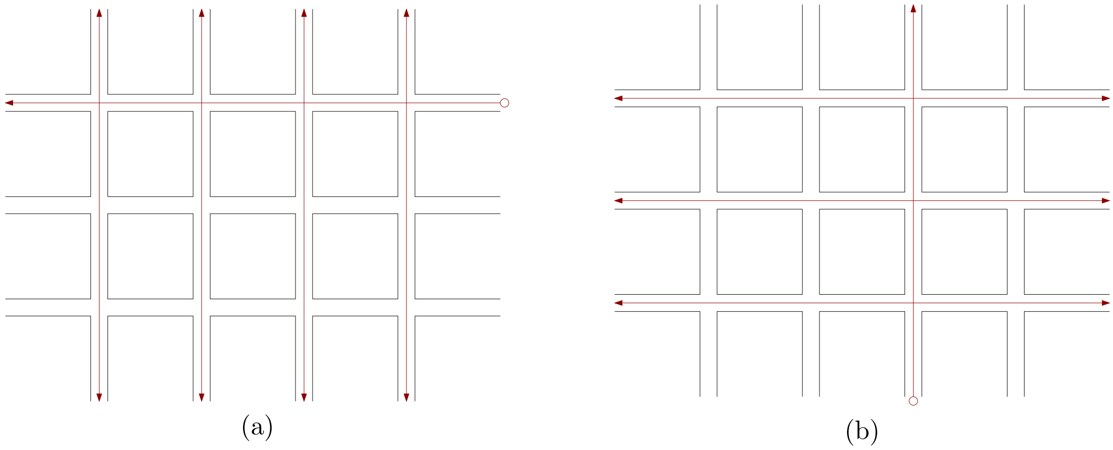


FIGURE 4.12: *The possible destinations for a vehicle approaching from the top easterly (horizontal) direction in a grid are displayed in (a), while the possible destinations for a vehicle approaching from the south in the vertical direction are displayed in (b).*

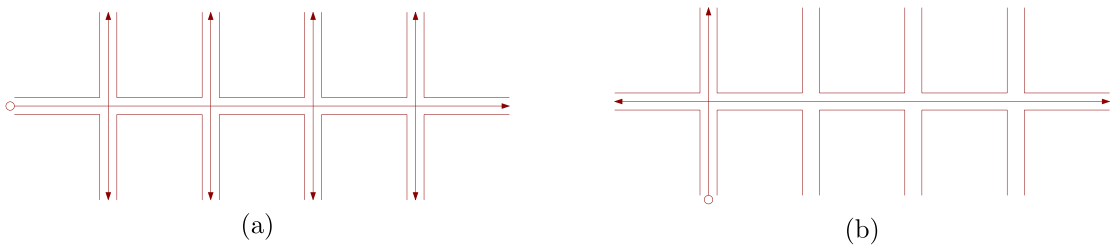


FIGURE 4.13: *The possible destinations for a vehicle approaching from a westerly direction in a four intersection corridor are displayed in (a), while the possible destinations for a vehicle approaching from the south in the vertical direction are displayed in (b).*

phase continues until either there are no more vehicles turning right, or until a maximum of 10 seconds has elapsed.

In the grid network, the probability of a vehicle turning left at a given intersection is 5%, while the probability of a vehicle turning right at an intersection is also 5%. For the corridor, the probability of a vehicle turning left or right at an intersection if it approaches from a horizontal direction is 2.5%, while it changes to 5% when a vehicle is generated to enter in a vertical direction.

4.3.3 Types of statistical analysis performed on model output data

For each scenario in which the algorithms are compared later in this dissertation, significant differences reported in the data are statistically significant at a 95% confidence level. First, an *analysis of variance* (ANOVA) is carried out in respect of each of the PMIs described in §4.1.4, indicating whether or not there is a significant difference between at least two of the algorithmic means of these PMIs. The test, however, only indicates *whether* there is at least one difference between two means, but does not specify *where* this difference occurs. If the ANOVA reveals that there is at least one significant difference between two means, a *post hoc* test is required to determine *where* this difference actually occurs. Unfortunately, most *post hoc* tests, however, assume homogeneity of variances, which is not always the case in the PMI data of §4.1.4.

Fisher's Least Significant Difference (LSD) test [85] and the *Games-Howell* test [19] are the two *post hoc* tests employed in this dissertation in respect of the statistical analysis of the difference in means of the data. After the ANOVA has been performed and has positively detected significant differences in the data, a Levene test [67] is carried out to determine whether the corresponding variances are significantly different from one another. If the variances are found not to be significantly different from one another, the LSD test is employed to find the location of these differences. If, however, the Levene test reveals that the variances are statistically different, the Games-Howell test is employed to determine where these differences lie.

The ANOVA, LSD, Levene and Games-Howell statistical tests are described in some detail in the remainder of this section.

The ANOVA test

The ANOVA test uses both the sum of squares between sets of data and sum of squares within sets of data to calculate whether there are significant differences in the data set means. The *sum of squares within groups* (SS_W) is given by

$$SS_W = \sum_{i=1}^{n}\sum_{j=1}^{m}(x_j - \bar{x}_i)^2, \tag{4.4}$$

where n denotes the number of data sets, m indicates the number of observations in each data set and $\bar{x}_i$ denotes the mean value for data set i. Similarly, the sum of squares between groups is given by

$$SS_B = m\sum_{i=1}^{n}(\bar{x}_i - \bar{x})^2, \tag{4.5}$$

where $\bar{x}$ additionally represents the average mean for the n data sets. The *mean square* (MS) is then calculated for both the SS_W and SS_B values, by dividing the sum of squares by the number of degrees of freedom. The *mean square within groups* (MS_W) and the *mean square between groups* (MS_B) are therefore given by

$$MS_W = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(x_j - \bar{x}_i)^2}{mn - n} \tag{4.6}$$

and

$$MS_B = \frac{m\sum_{i=1}^{n}(\bar{x}_i - \bar{x})^2}{n - 1}, \tag{4.7}$$

respectively. Once these two values have been obtained, the ratio between them, denoted by F, is calculated as

$$F(n-1, mn-n) = \frac{MS_B}{MS_W}. \tag{4.8}$$

The critical value in the F distribution table (f) is then compared at a significance value of $p < 0.05$ with the calculated value $F(n-1, mn-n)$ in (4.8). If $F(n-1, mn-n) < f$, there is no statistically significant difference between the means at a 95% confidence level. If, however, $F(n-1, mn-n) > f$, then there is a statistically significant difference between at least two of the means, necessitating a *post hoc* test to find where this difference occurs.

The Levene test

The Levene test is used to assess whether the variances of two or more data sets are significantly different. It requires the determination of two variables, F_L and $F(\alpha, k-1, N-k)$, where $F(\alpha, k-1, N-k)$ is obtained from the F-ratio table at a significance level of α. F_L is given by

$$F_L = \frac{(N-k)\sum_{i=1}^{k} N_i(\bar{x}_i - \bar{x})^2}{(k-1)\sum_{i=1}^{k}\sum_{j=1}^{N_i}(|x_{ij} - \bar{x}_i| - \bar{x}_i)^2}, \tag{4.9}$$

where N is the total number of data points in k sets, N_i is the number of data points in set i, $\bar{x}_i$ is the mean of set i, $\bar{x}$ is the mean over all data points in all k sets and x_{ij} is data point i from data set j. If

$$F_L \geq F(k-1, N-k), \tag{4.10}$$

the variances between at least two of the data sets are significantly different and the Games-Howell test is carried out. If, on the other hand, (4.10) does not hold, there are no significant differences between any of the data sets at a $(100-\alpha)$ level of confidence and the LSD test is carried out.

The Fisher LSD *post hoc* test

Fisher's LSD *post hoc* test is a very powerful parametric statistical test. It has, however, been criticised due to the belief that it does not protect against inflated type 1 error rates, although this is only the case when the number of data sets being compared is more than three [27]. According to Kidd [39], however, Fisher's LSD test is appropriate in large designs with many *post hoc* comparisons, so long as the *practical significance*[1] is also taken into account.

The statistic of the LSD test at a 95% level of significance is given by

$$LSD_{A,B} = t_{0.05/2, DF_W}\sqrt{MS_W(1/m_A + 1/m_B)}, \tag{4.11}$$

where A and B represent two different data sets being compared, m_A and m_b denote the number of data points in set A and set B, respectively, and DF_W denotes the number of degrees of freedom within these sets. The means $\bar{x}_A$ and $\bar{x}_B$ of respectively the sets A and B are calculated and the absolute value of their difference is determined. If

$$|\bar{x}_A - \bar{x}_B| \geq LSD_{A,B}, \tag{4.12}$$

then a significant difference exists between these two means at a 95% level of confidence. If not, there is no significant difference at a 95% level of confidence. This test has to be repeated for all $\binom{n}{2}$ pairs of data sets.

The Games-Howell *post hoc* test

The Games-Howell *post hoc* test [35, 34] is a non-parametric test that is recommended if the sample sizes are unequal or if the homogeneity of variances assumption required for Fisher's

[1]Practical significance refers to evaluating whether statistically significant differences are large enough to be of value in a practical sense. Consider two algorithms that, after a number of simulation runs, are found to yield significantly different results with respect to mean delay times, differing by only 0.3 seconds. While it may have been proven that these means are statistically different, it is clear that this difference is not of practical importance.

LSD test is violated [18]. It has also been referred to as one of the most robust modern methods of *post hoc* testing, and is a more conservative test than the majority of other *post hoc* tests [33]. This test makes use of Welch's degrees of freedom (from Welch's t-test[2]), and the studentised range distribution[3], denoted by q, and is given by

$$|\bar{x}_A - \bar{x}_B| > q_{\sigma,n,df}, \tag{4.13}$$

where

$$\sigma = \sqrt{\frac{1}{2}\left(\frac{s_A^2}{m_A} + \frac{s_B^2}{m_B}\right)} \tag{4.14}$$

and

$$df = \frac{\left(\frac{s_A^2}{m_A} + \frac{s_B^2}{m_B}\right)^2}{\frac{\left(\frac{s_A^2}{m_A}\right)^2}{m_A-1} + \frac{\left(\frac{s_B^2}{m_B}\right)^2}{m_B-1}}. \tag{4.15}$$

In (4.14)–(4.15), n is the number of data sets, m_A and m_B are the number of observations in data sets A and B, respectively, and s_A and s_B are the standard deviations of data sets A and B, respectively. If (4.13) holds, there is a significant difference between the two means of data sets A and B at a 95% level of confidence. If, on the other hand, (4.13) is not satisfied, then there is no significant difference between the two means at a 95% level of confidence.

P-values in hypothesis testing

Fixed significance level testing in inferential statistics involves determining whether or not a null-hypothesis H_0 should be rejected at a chosen level of significance α. This test returns a *p-value* that represents the probability that a test statistic obtains a value that is at least as extreme as the observed value, given that the null-hypothesis is true. The p-value is therefore the smallest level of significance for which H_0 should be rejected. An example of how the p-value is calculated in the Fisher LSD test is given for the hypotheses

$$H_0 : |\bar{x}_A - \bar{x}_B| = 0 \qquad H_1 : |\bar{x}_A - \bar{x}_B| \neq 0. \tag{4.16}$$

In this case, the p-value is calculated as

$$1 - P\left(-\frac{|\bar{x}_A - \bar{x}_B|}{\sqrt{MS_W(1/m_A + 1/m_B)}} < t_{0.05/2,DF_W} < \frac{|\bar{x}_A - \bar{x}_B|}{\sqrt{MS_W(1/m_A + 1/m_B)}} \right). \tag{4.17}$$

Once the p-value has been calculated, it is compared to the level of significance α. If the p-value is smaller than α, the null hypothesis H_0 is rejected, while if the p-value exceeds α, H_0 is not rejected. The p-value may therefore be interpreted as the probability of incorrectly rejecting H_0 (*i.e.* making a so-called Type I error). The corresponding p-values for the ANOVA, Levene and Games-Howell tests are all determined similarly, but utilising the appropriate probability distributions in (4.17) for each test.

[2] Welch's test is a two-sample location test that is used for testing whether two means from different populations are equal. This test does not assume homogeneity of variance, but does assume normality of data.

[3] A distribution used for estimating the range of a normally distributed population, when the standard deviation is unknown and the population is considered small

4.4 Summary

This chapter opened in §4.1 with a description of the various entities involved in building the simulation model used as a test bed later in this dissertation to compare the traffic signal control algorithms. These entities included creating the road network, implementing the traffic signals and generating vehicles. A number of verification and validation techniques that were applied to the model were described in §4.2, with a focus on the validation front where a comparison was made between the model output and the corresponding output of a real system. The nature of the experimental design was described in §4.3. These included a discussion on the determination of a suitable simulation warm-up period, the specification of parameters in the simulation framework, and the statistical tests that are to be used later in this dissertation to analyse the model output data.

CHAPTER 5

Algorithm implementation and improvements

Contents

This chapter serves to describe exactly how a fixed-time control algorithm and five of the self-organising algorithms reviewed in Chapter 2 were interpreted, implemented, and executed in order to facilitate a comparison of their relative effectiveness. Based on numerical experiments, a number of shortcomings are identified in some of the algorithms and the effects of these shortcomings are noted. Various algorithmic improvements are subsequently proposed along with supporting reasons for these suggestions. A comparison is finally carried out in respect of the relative performances of the algorithms after having implemented these changes. The results pertain to simulation runs in the context of a 3×4 grid of intersections, and are represented by means of box plots and appropriate statistical analyses for both lighter and heavier traffic conditions.

The six algorithms considered in this chapter include a fixed-time control strategy, the three recent self-organising algorithms proposed by Einhorn [16] (I-TSCA, O-TSCA, Hybrid) as well as the earlier algorithms of Gershenson and Rosenblueth [22] (referred to as Gersh) and of Lämmer and Helbing [46] (referred to as LH). As stated in Chapter 2, the algorithm by Cesme [12] is excluded from the analysis as it consists of an extensive list of rules applicable for a range of different scenarios, including *transit signal priority*, which is not taken into account in the simulation model. This control strategy, as well as the algorithm by Xie *et al.* [87], furthermore assumes instantaneous vehicle detection by means of detectors embedded within the road surface. Both of these algorithms employ the distance between detectors as parameters. They are therefore not included in the algorithmic comparison carried out in this chapter as it is assumed that algorithms making use of radar detection will perform more effectively within the paradigm of self-organisation, given the more detailed information required by algorithms in this paradigm.

5.1 Implementation of the algorithms

In this section, a fixed-time control strategy is described in which the cycle lengths and offsets are calculated. The parameters of the five self-organising algorithms mentioned above are defined, and state charts defining the logic of the algorithms are provided. Detailed descriptions are also included of how the I-TSCA, the O-TSCA and the Hybrid algorithm were interpreted and implemented. Since the original specification of certain aspects of these three algorithms were found to be unclear during their implementation, this section serves to describe exactly how these algorithms are assumed to function in this dissertation. For the sake of completeness, similar descriptions and state charts are also provided for the Gersh algorithm and the LH algorithm.

5.1.1 A fixed-time control strategy

The fixed-time control algorithm makes use of a fixed-time signal control strategy and is from here on referred to as *Fixed*. If a number of intersections are located relatively near to one another, it is desirable to coordinate their signal timings in such a manner that vehicles receive green signals as they reach consecutive intersections when travelling through a transportation network. This is done through the determination of a suitable cycle length, an offset time and green times for various movements through the intersections.

In this dissertation, the signal timings at each intersection are implemented in a cycle of length C, measured in seconds. The value of C is calculated by utilising the optimal cycle length formula developed by Webster [83] which aims to minimise vehicle delays when considering random vehicle arrivals. This formula is given by

$$C_{opt} = \frac{1.5L + 5}{1.0 - Y}, \tag{5.1}$$

where L is the lost time per cycle (the sum of the setup times in one cycle) and Y is the sum of the critical lane volume divided by the saturation flow for each phase. Exclusive right-turning phases are not incorporated in Fixed as the Highway Capacity Manual [32] states that this is only recommended if the turn volume is more than 100 vehicles per hour. The cycle therefore comprises two green phases and two setup times, resulting in a lost time per cycle of $L = 10$ seconds, while the saturation flow is taken as 1 800 vehicles per hour. Light traffic conditions result in a flow of 600 vehicles per hour, and thus the cycle length for light traffic conditions is 30 seconds. After subtracting the setup time from this value, the green times for each of the green phases are calculated to be 10 seconds each (an equal demand is assumed in both directions). The traffic flow under heavy traffic conditions is 1 200 vehicles per hour, and so the cycle length becomes 60 seconds in this case. Once the setup time is subtracted, the two green phases each last 25 seconds.

The distance between neighbouring intersections is assumed to be 385 metres, while the average speed of vehicles is taken as 16.67m/s. The offset is calculated by obtaining the time in seconds required by a vehicle to travel from one intersection to the next. This is calculated by dividing the distance of 385 metres by the speed of 16.67m/s to obtain a time of 23 seconds. An additional two seconds are added to account for reaching the desired speed if there is a vehicle queue or a slower vehicle travelling below 60km/h, resulting in an offset time of 25 seconds. The intersections are coordinated in such a way that the wave of uninterrupted traffic flows in a west-to-east direction and a north-to-south direction.

5.1.2 The O-TSCA of Einhorn

During implementation of the O-TSCA, the author realised that no indication was given in [16] of the length of road considered during the calculation of phase pressures. Einhorn [16], who proposed the algorithm, was contacted and it was confirmed that the road length considered was 275 metres, as that is the furthest distance over which the *SmartSensor Advance Extended Range* [82] detection unit depicted in Figure 1.4 can detect vehicles.

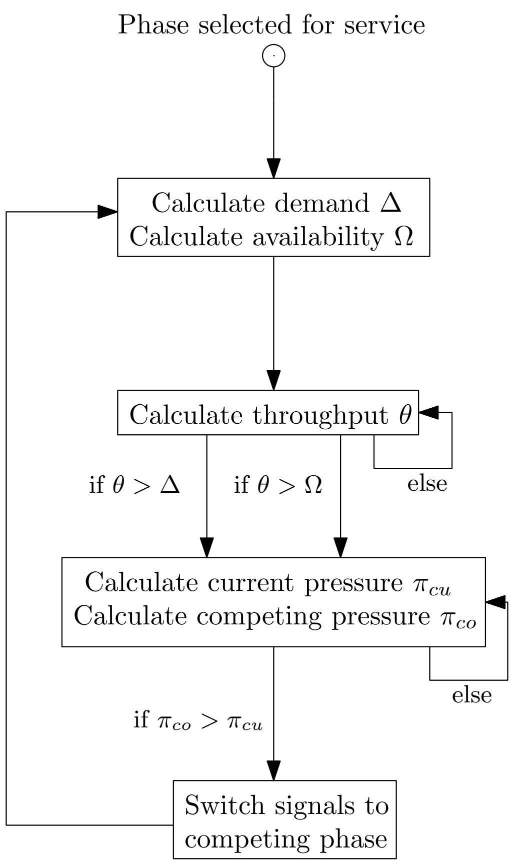


FIGURE 5.1: *State chart of the O-TSCA of Einhorn [16].*

In Figure 5.1, the logic of the O-TSCA is represented in state chart form. A phase is selected for service and the current demand Δ, as well as the current availability Ω, along the 275 metre stretch of road is stored in the first state. In the second state, the intersection throughput is updated periodically (every 0.5 seconds in the model implementation of this dissertation). Once the throughput exceeds either the demand or availability, the pressures of both the current and competing traffic flows are calculated and compared in the third state. If the competing traffic flow achieves a larger pressure than that of the current flow receiving service, the signal is changed. On the other hand, if the current traffic flow has a larger pressure than that of the competing flow, the pressures are re-evaluated every 0.5 seconds until one of the former conditions is satisfied in which case the signals change.

5.1.3 The I-TSCA of Einhorn

A similar lack of clarity was discovered during implementation of the I-TSCA. Calculation of the extended required green time was not properly specified by Einhorn [16]. It was only stated that a phase may be extended if it results in the lowest total cost. According to the description of the algorithm, the required green time is the time taken to clear the shortest queue or else, if there is no queue, the time taken for the closest vehicle to reach the intersection. The decision as to whether or not signals switch, occurs right before the end of the phase currently receiving green time. Therefore, at this point in time, all the vehicles along the approach currently receiving service belong to the set $Q_i(t)$ (the set of all currently queued vehicles as well as those that are predicted to become queued). If all these vehicles belong to $Q_i(t)$, then the extended green time is the shortest expected time taken for the last vehicle in one of the approach lanes to reach the intersection. The algorithm was implemented this way initially, but yielded poor results and did not result in traffic flows as described in the literature. It was therefore assumed that when calculating the extended green time of a phase, the green time should be the expected time taken by the vehicle closest to the intersection to reach it. This leads to signals switching fairly often, a phenomenon reported by Einhorn [16], to be a prevalent feature of the I-TSCA.

The logic of the I-TSCA is apparent in the state chart in Figure 5.2. Once a phase has been selected for service, the required green time for that phase is calculated in the first state and this green time is assigned in the second state. Just before the end of the green time (if 0.5 seconds or less is left of green time in the model implementation of this dissertation), the extended green time for the current phase is calculated in the third state according to the logic described above. The total delay associated with extending the green signal and switching the signal is calculated and compared in the fourth state. If the predicted delay of extending the green time is less than that of switching signals, the green time is extended; otherwise the signals change.

5.1.4 The Hybrid algorithm of Einhorn

The Hybrid algorithm of Einhorn [16] was implemented according to the above logic and assumptions made in respect of the O-TSCA and the I-TSCA. The I-TSCA assigns specific green times in advance, while the O-TSCA assigns an infinite green time and switches signals once certain conditions are met. As a result, the I-TSCA and the O-TSCA are implemented slightly differently in the Hybrid algorithm.

The logic of the Hybrid algorithm is presented in the form of a state chart in Figure 5.3. Once a phase has been selected for service, both the I-TSCA and the O-TSCA are executed concurrently in the first state, while the amount of green time is decided upon by the I-TSCA. Since the I-TSCA is associated with a faster signal change, the Hybrid waits for the I-TSCA to

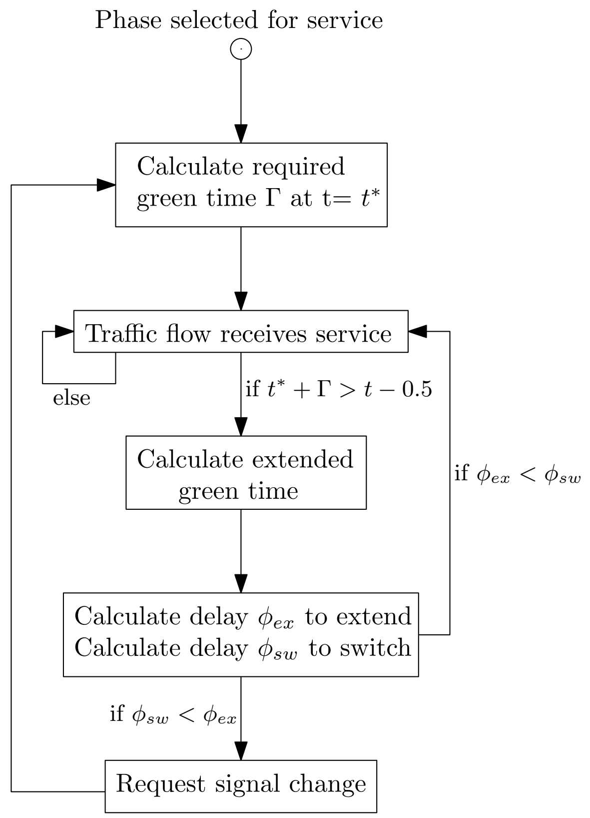


FIGURE 5.2: *State chart of the I-TSCA of Einhorn [16].*

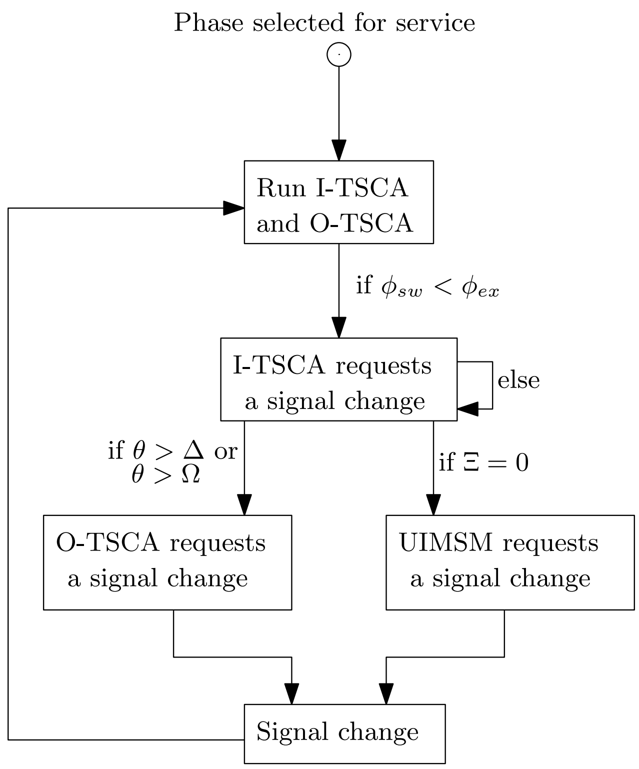


FIGURE 5.3: *State chart of the Hybrid algorithm of Einhorn [16].*

request a signal change in the second state. Once the I-TSCA has requested the signal change (*i.e.* an extended green time was calculated and rejected by the algorithm), the I-TSCA allows the extended green time (which was rejected) to begin, until either the IUMSM or the O-TSCA also requests a signal change, at which point in time the signals change.

5.1.5 The algorithm of Gershenson and Rosenblueth

The algorithm by Gershenson and Rosenblueth [22] requires as input parameters the threshold φ_I, the minimum green time μ_I, the distances d, r and e from the intersection (as illustrated in Figure 2.1), and the maximum number of vehicles s that may extend a green signal. Since it is assumed in [22] that traffic only moves in one direction along a roadway, which is not the case in the simulation model of this dissertation, the threshold values recommended by Gershenson and Rosenblueth [22] are doubled. In order to account for a second lane in each direction, the threshold value is doubled again to obtain the threshold value of 53.33. The minimum green time is furthermore not set to the recommended value of 3.33 seconds, since this time interval is so short that vehicles have barely started moving when this interval elapses, resulting in a situation where there are generally more than s vehicles within a distance r from the intersection (which does not prevent signals from switching). This goes on to allow rule 1 to be evaluated and this rule regularly allows the switching of signals. The minimum green time was therefore taken as 7 seconds instead, according to recommended minimum green times in the Traffic Signal Timing Manual of the U.S. Department of Transportation [42]. The distances d, r and e were taken as the values recommended in [22] of 50, 25 and 10 metres, respectively.

Since turning is assumed in [22] not to be allowed, the right turning lane is ignored when determining whether there are vehicles within a distance r from the intersection, because the signals will not switch if there is such a vehicle waiting to turn.

It was found that Gersh performs comparatively well under lighter traffic conditions, but poorly under heavier traffic conditions. Through close inspection of execution of this algorithm within the simulation model of Chapter 4 it became apparent that the cause of this deficiency lies in rules 3 and 4. Under both lighter and heavier traffic conditions, rule 3 typically prevents the signals from changing since it is rare that more than $s = 2$ vehicles are within $r = 25$ metres of the intersection. The absence of vehicles within a distance $d = 50$ metres from the intersection of an approach currently receiving service together with the presence of at least one vehicle within 50 metres in an opposing direction leads to the enforcement of rule 4, causing the signals to change. This general behaviour of the algorithm is demonstrated in Figure 5.4, typically cycling through rules 6–3 until ultimately switching signals as a result of rule 4.

While this behaviour is efficient under lighter traffic conditions, it is not the case under heavier traffic conditions. It is common that there are substantial gaps between consecutive vehicles under lighter traffic conditions, causing signals to change reasonably often due to rule 4, which

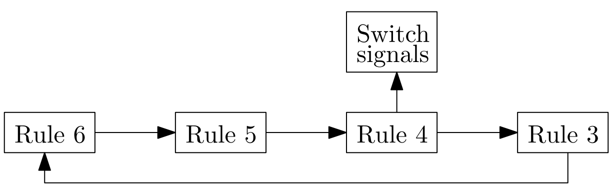


Figure 5.4: *An example of typical behaviour displayed by Gersh in respect of the rules implemented in the algorithm.*

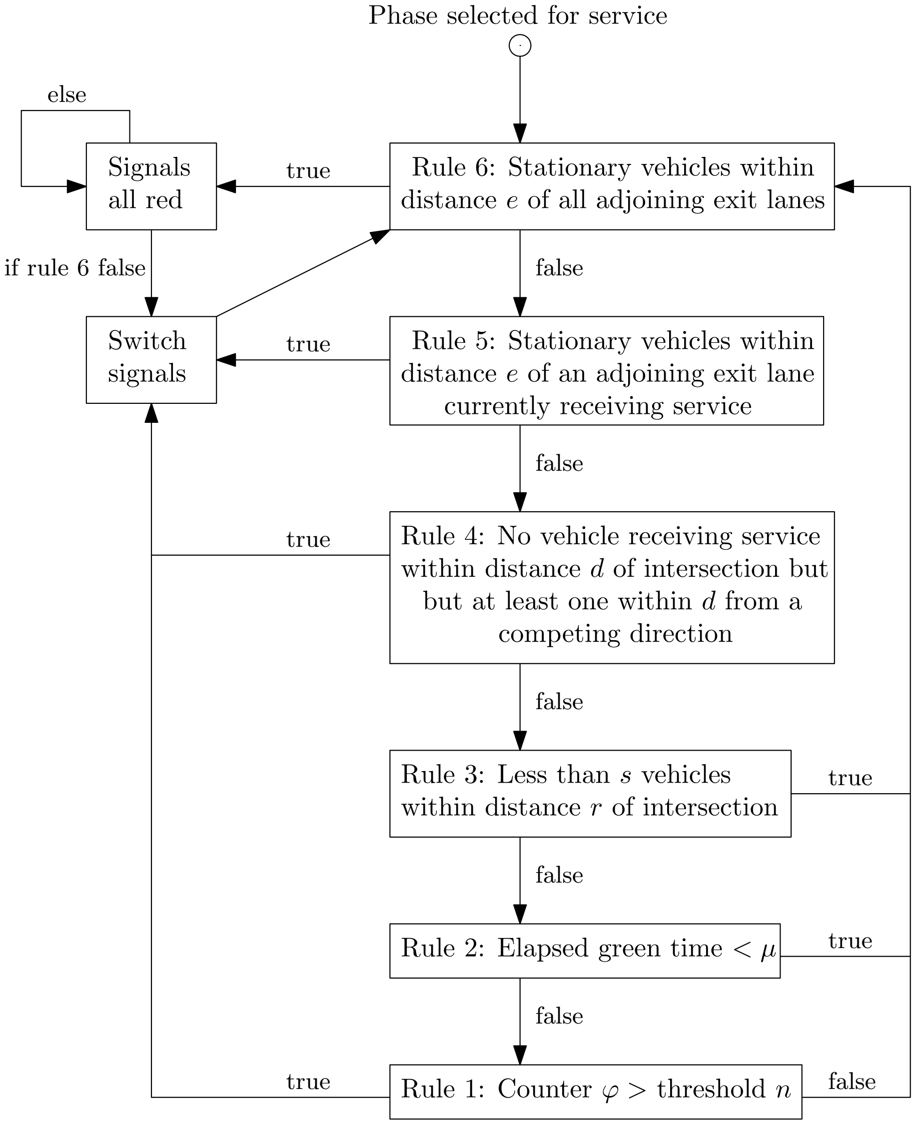


FIGURE 5.5: *State chart of the algorithm by Gershenson and Rosenblueth [22].*

is desirable. These gaps are, however, less frequent as the traffic density increases, and as a result rule 4 becomes more resistant to changing signals. This results in longer green times and increased delays experienced by vehicles.

The purpose of rule 3 is to prevent signals from switching if there are s or fewer vehicles within a distance r from the intersection. This serves to avoid separating the tail of a platoon under heavy traffic conditions. The rule does not, however, induce this intended behaviour in the simulation model of Chapter 4, since under heavy traffic conditions (if vehicles are travelling at speed) three or more vehicles are rarely within a distance r from the intersection, unless a vehicle is slowing down to turn left[1]. In order to ensure that the rule functions the way it was intended to function, it is suggested for future work that certain parameter values are varied, such as the distance r as well as the number of vehicles s.

A state chart representing the logic of the algorithm is shown in Figure 5.5. Once a phase is assigned service, the algorithm evaluates rule 6. If the criterion of this rule is satisfied, all signals are switched to red until it is no longer satisfied. On the other hand, if it is not satisfied, rule 5 is considered. If the criterion of rule 5 is satisfied, signals switch to service another traffic flow, while if the criterion is not satisfied, rule 4 is evaluated. Similarly, if the criterion of rule 4 is satisfied, the signals switch. If the criterion of rule 4 is not satisfied, rule 3 is evaluated. If the criterion of rule 3 is satisfied, signals are not switched. If, however, the criterion of rule 3 is not met, the minimum time is evaluated in rule 2. If the minimum time exceeds the threshold, rule 1 is assessed. If the threshold is not exceeded, the signals are not changed and rule 6 is assessed. Finally, if the criterion of rule 1 is satisfied, the signals are switched. Otherwise, the signals do not change and rule 6 is evaluated again. The process is repeated.

5.1.6 The algorithm of Lämmer and Helbing

The algorithm by Lämmer and Helbing [46] makes use of two parameters, namely Z (which is set to 90 seconds) and Z^{max} (which is set to 120 seconds) as is in the example in [47].

A state chart representing the logic of the algorithm by Lämmer and Helbing is shown in Figure 5.6. The first step in this algorithm is determining whether or not the queue lengths are unstable. If this is the case, the stabilisation strategy is enforced until the queues are no longer unstable (until $|\text{queue}| \leq n_{crit}$). If, however, the queue lengths are stable, the optimisation strategy is executed, whereby the priority indices for both the current and competing phase are calculated and compared. If the current phase has a larger priority index, the indices continue to be recalculated until the competing phase has a larger priority index, in which case the signals are switched.

5.2 Simulation results returned by the algorithms as-is

The results obtained from simulations involving the fixed-time control strategy and the five self-organising algorithms described in §5.1 are presented in this section and are interpreted through the use of box plots and tables indicating whether or not differences exist between the PMIs for each pair of algorithms at a 5% level of significance. These results are reported for the algorithms under both lighter and heavier traffic conditions, and are also compared with the results reported by Einhorn [16]. The PMIs considered in this specific comparison are the

[1]One of the assumptions made in this algorithm is that there is no turning at intersections. This is another indication that rule 3 is not functioning as expected.

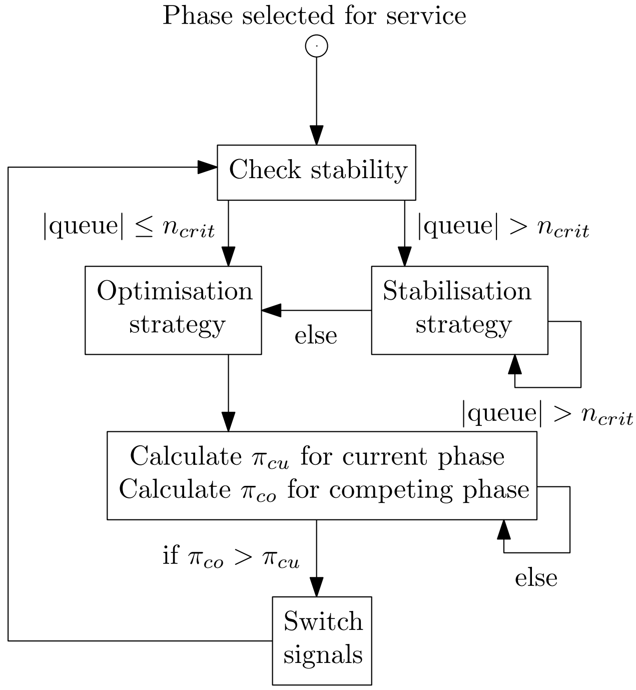


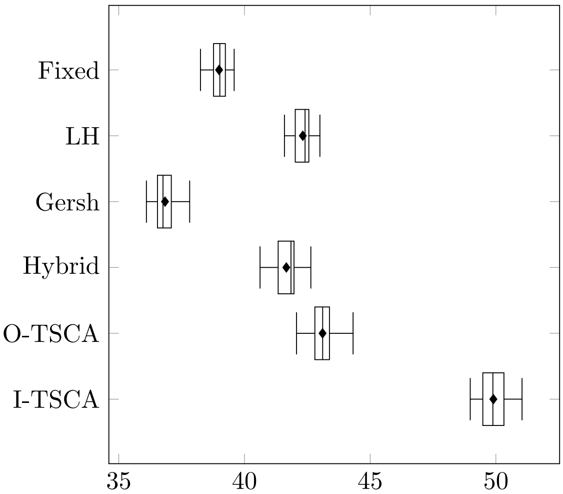
FIGURE 5.6: *State chart of the algorithm by Lämmer and Helbing [46].*

mean delay time, the normalised mean delay time, the mean number of stops and the normalised mean number of stops. The other two PMIs mentioned in §4.1.4 were not a part of the study by Einhorn [16] — their values therefore cannot be compared to the results report by Einhorn [16]. Of course, the actual PMI values cannot be compared directly as different simulation models were used, incorporating different vehicle lengths, road lengths and signal phases. It is noted, however, that both the results reported in this section and those reported by Einhorn [16] were obtained based on simulations in the context of a 3×4 grid of equally spaced intersections, with identical arrival rates, vehicle speeds and turning probabilities. Due to these similarities, the results in this experiment are expected to be rather similar to those reported by Einhorn [16].

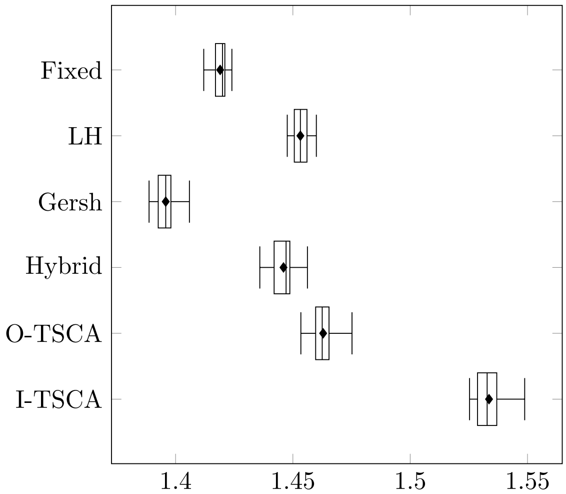
5.2.1 Results under lighter traffic conditions ($\lambda = 10$ vehicles per minute)

The ANOVA column in Table 5.1 indicates that for all six PMIs there are statistical differences between the means returned by the algorithms at a 5% level of significance. Furthermore, the results of the Levene tests revealed that the variance of the mean samples are not statistically distinguishable for PMI 6, and so the Fisher LSD *post hoc* test was used in respect of this PMI in order to determine between which pairs of algorithmic outputs statistical differences are discernible. The variance of means for PMI 1, PMI 2, PMI 3, PMI 4 and PMI 5 are, however, statistically different at a 5% level of significance and therefore the Games-Howell *post hoc* test was performed in respect of these five PMIs for this purpose.

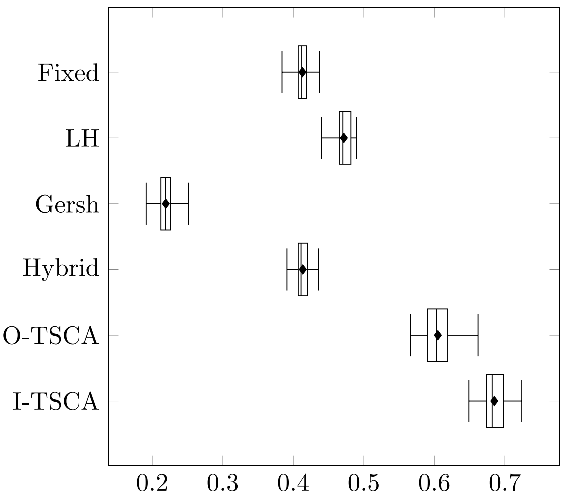
For each of the PMIs, a similar trend emerges with respect to the order of the comparative performance of the algorithms, which is clear from box plots of these performances, shown for lighter traffic conditions (an arrival rate of $\lambda = 10$ vehicles per minute) in Figure 5.7. Gersh was found to be the algorithm that performs the best overall, achieving the most favourable outcome for each of the six PMIs. Fixed performed the second most effectively overall, followed by Hybrid. The O-TSCA was the second worst performing algorithm in respect of each PMI,



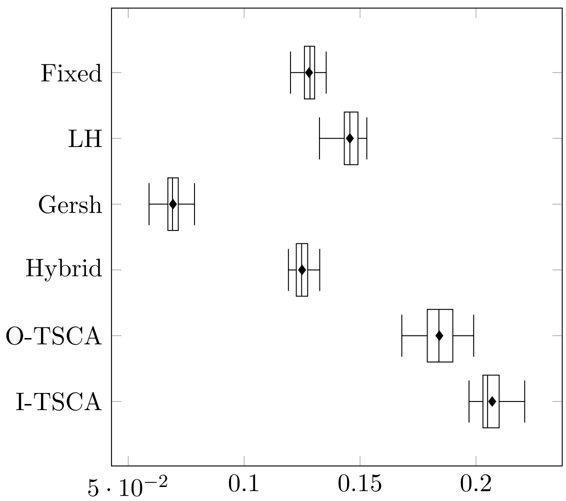
(a) *Mean delay*



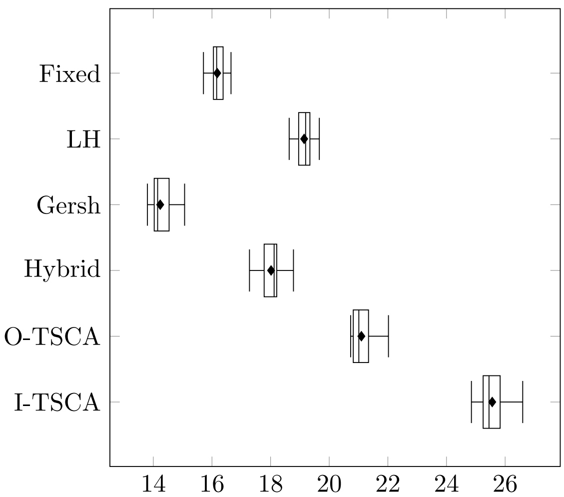
(b) *Normalised mean delay*



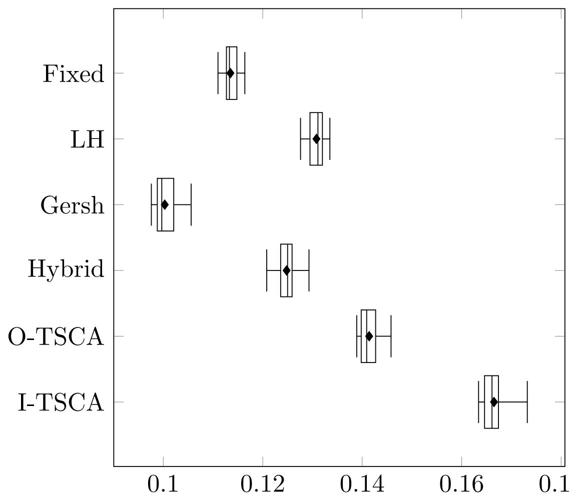
(c) *Mean number of stops*



(d) *Normalised mean number of stops*



(e) *Mean time spent travelling under 10km/h*



(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 5.7: *PMI results for the six algorithms of §5.1 in the context of a 3×4 grid of intersections under lighter traffic conditions.*

	Mean value						p-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	ANOVA	Levene's Test
1	49.90	43.11	41.67	36.84	42.32	38.99	$<1 \times 10^{-17}$	4.43×10^{-2}
2	1.534	1.463	1.446	1.396	1.453	1.419	$<1 \times 10^{-17}$	4.68×10^{-2}
3	0.685	0.605	0.413	0.219	0.472	0.413	$<1 \times 10^{-17}$	9.25×10^{-6}
4	0.207	0.184	0.125	0.069	0.146	0.128	$<1 \times 10^{-17}$	3.68×10^{-6}
5	25.56	21.10	18.02	14.24	19.15	16.18	$<1 \times 10^{-17}$	7.73×10^{-3}
6	0.1665	0.1414	0.1248	0.1003	0.1308	0.1135	$<1 \times 10^{-17}$	2.05×10^{-1}

TABLE 5.1: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under lighter traffic conditions in a 3×4 grid of intersections. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean delay						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	2.51×10^{-12}	3.72×10^{-12}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
O-TSCA		—	1.48×10^{-11}	1.45×10^{-11}	4.15×10^{-9}	4.12×10^{-12}
Hybrid			—	1.42×10^{-11}	5.29×10^{-7}	2.89×10^{-12}
Gersh				—	$<1.48 \times 10^{-12}$	7.50×10^{-12}
LH					—	1.37×10^{-11}
Fixed						—
Mean	49.90	43.11	41.67	36.84	42.32	38.99

TABLE 5.2: *Differences in respect of the mean delay obtained for each of the original self-organising algorithms under lighter traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean delay						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	1.41×10^{-12}	3.24×10^{-12}	6.99×10^{-13}	2.86×10^{-13}	$<1 \times 10^{-17}$
O-TSCA		—	1.48×10^{-11}	1.49×10^{-11}	8.98×10^{-13}	$<1 \times 10^{-17}$
Hybrid			—	1.47×10^{-11}	5.77×10^{-8}	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
LH					—	1.48×10^{-11}
Fixed						—
Mean	1.534	1.463	1.446	1.396	1.453	1.419

TABLE 5.3: *Differences in respect of the normalised mean delay obtained for each of the original self-organising algorithms under lighter traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
O-TSCA		—	2.14×10^{-13}	3.78×10^{-13}	8.98×10^{-13}	4.68×10^{-13}
Hybrid			—	1.48×10^{-11}	1.30×10^{-11}	9.99×10^{-1}
Gersh				—	1.39×10^{-11}	1.49×10^{-11}
LH					—	1.42×10^{-11}
Fixed						—
Mean	0.685	0.605	0.413	0.219	0.472	0.413

TABLE 5.4: *Differences in respect of the mean number of stops obtained for each of the original self-organising algorithms under lighter traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	7.28×10^{-13}	$<1 \times 10^{-17}$
O-TSCA		—	1.74×10^{-13}	8.37×10^{-13}	1.01×10^{-12}	6.87×10^{-13}
Hybrid			—	1.33×10^{-11}	7.18×10^{-12}	2.58×10^{-2}
Gersh				—	1.27×10^{-11}	1.48×10^{-11}
LH					—	1.18×10^{-11}
Fixed						—
Mean	0.207	0.184	0.125	0.069	0.146	0.128

TABLE 5.5: *Differences in respect of the normalised mean number of stops obtained for each of the original self-organising algorithms under lighter traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean time spent travelling under 10km/h						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	8.06×10^{-13}
O-TSCA		—	1.48×10^{-11}	1.46×10^{-11}	2.85×10^{-12}	$<1 \times 10^{-17}$
Hybrid			—	1.49×10^{-11}	4.86×10^{-12}	4.51×10^{-13}
Gersh				—	5.78×10^{-12}	1.30×10^{-12}
LH					—	1.43×10^{-11}
Fixed						—
Mean	25.56	21.10	18.02	14.24	19.15	16.18

TABLE 5.6: *Differences in respect of the mean time spent travelling under 10km/h obtained for each of the original self-organising algorithms under lighter traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Normalised mean time spent travelling under 10km/h						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
O-TSCA		—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Hybrid			—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$
Fixed						—
Mean	0.1665	0.1414	0.1248	0.1003	0.1308	0.1135

TABLE 5.7: *Differences in respect of the normalised mean time spent travelling under 10km/h obtained for each of the original self-organising algorithms under lighter traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

followed by I-TSCA which was consistently the worst performing algorithm in terms of each of the PMIs at a 5% level of significance. The presence of significant differences between the results returned by the various algorithms is elucidated in the Fisher LSD and Games-Howell test results in Tables 5.2–5.7.

Gersh achieved a mean delay time of 36.84 seconds, as shown in Figure 5.7(a) and Table 5.2, a 4.83 second improvement over the next best performing algorithm in respect of this PMI. A similar result was found in terms of mean time spent under 10km/h, where Gersh achieved a value of 14.24 seconds, outperforming the next best algorithm in respect of this PMI by 3.78 seconds, as can be seen in Figure 5.7(e) and Table 5.6.

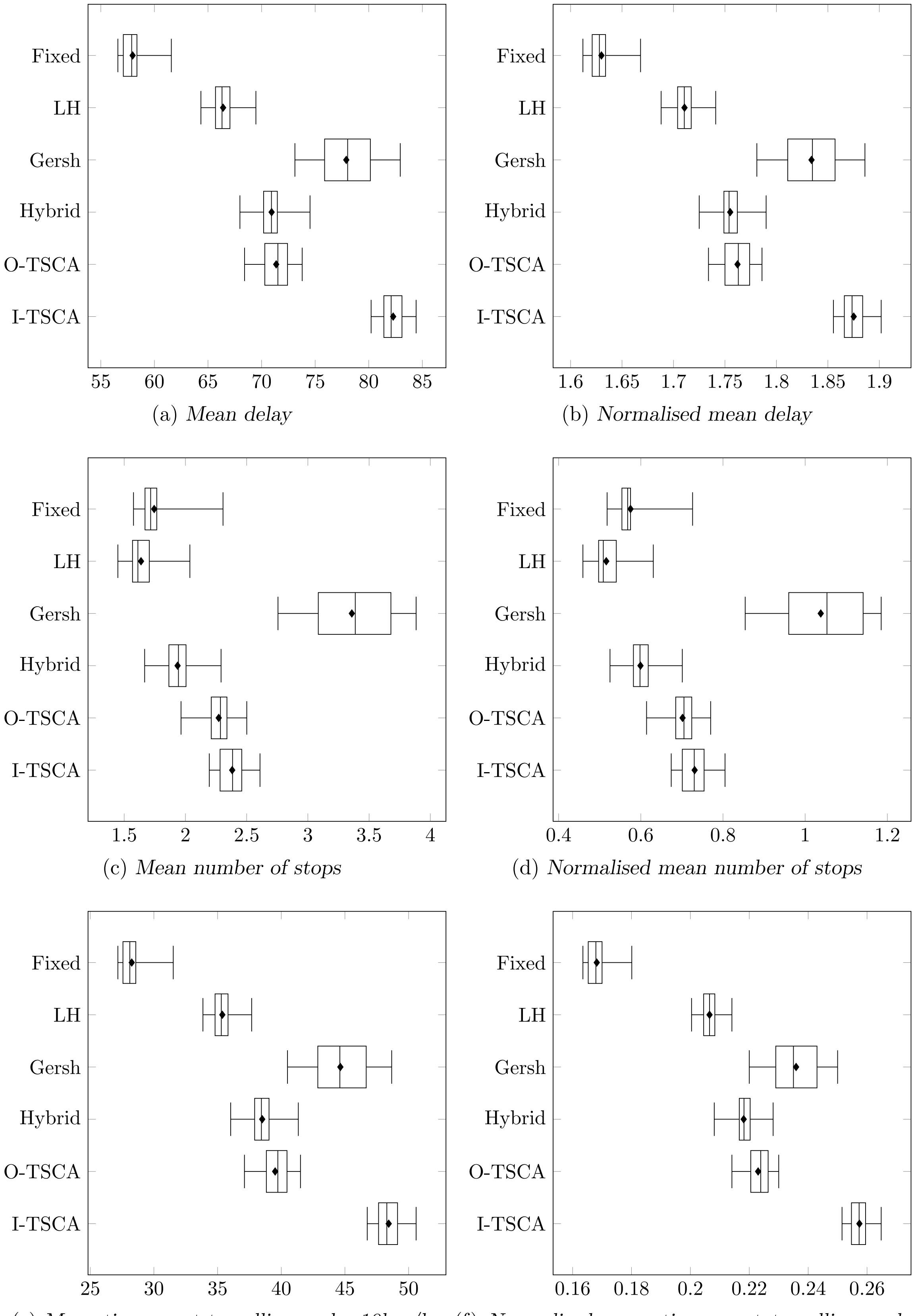
The very short green times allocated by the I-TSCA (as short as 3.33 seconds) is the principal cause for the poor performance of the algorithm, often only allowing the front vehicle row to make it through the intersection before a signal change is initiated. The O-TSCA, on the other hand, performed poorly for a contrasting reason: excessive green time. It was clear from observing the working of the O-TSCA in the simulation model that the signals switched too infrequently, often causing vehicles to wait unnecessarily long at intersections. Hybrid was able to improve upon both the I-TSCA and O-TSCA, and this improvement is attributed to the IUMSM, maximising intersection utilisation by switching signals according to the more suitable algorithm at the current time.

A number of differences were found between the results reported in Figure 5.7 and those reported by Einhorn [16]. One of the most notable differences is the finding in terms of the best performing algorithm in respect of the mean number of stops. According to Einhorn [16], the O-TSCA achieves the smallest number of stops under lighter traffic conditions, followed by Hybrid, whereas it was found in this study that the O-TSCA was relatively ineffective at preventing vehicle stops. In terms of mean delay time, it was further found by Einhorn that Hybrid was the most effective algorithm, followed by Gersh, while LH was the worst performing algorithm. In contrast to this finding, it is reported here that Gersh is the best performing algorithm under lighter traffic conditions, followed by LH. It was also reported by Einhorn [16] that Hybrid was the best performing algorithm in terms of normalised mean delay time, significantly outperforming the second best algorithm, Gersh. The opposite is the case for the results in Figure 5.7. Finally, LH was the worst performing algorithm overall according to Einhorn [16], contradicting the results of Figure 5.7, which indicate that I-TSCA performs the worst overall under lighter traffic conditions.

5.2.2 Results under heavier traffic conditions ($\lambda = 20$ vehicles per minute)

Once again the ANOVA column in Table 6.22 indicates that for all six PMIs there are statistical differences between the means returned by the algorithms at a 5% level of significance. Furthermore, the results of the Levene tests revealed that the variances of the mean samples are statistically different for all six of the PMIs. Therefore the Games-Howell *post hoc* test was performed in respect of all the PMIs in order to determine between which pairs of algorithmic outputs statistical differences are discernible.

Interestingly, it was found that under heavier traffic conditions ($\lambda = 20$ vehicles per minute), the performance of Gersh worsened dramatically in comparison to the other four algorithms, which all worsened by a similar margin. The variance in the PMIs associated with Gersh were also significantly larger than it was under lighter traffic conditions, indicating that there are inconsistencies in this algorithm under heavier traffic conditions.



(a) *Mean delay*

(b) *Normalised mean delay*

(c) *Mean number of stops*

(d) *Normalised mean number of stops*

(e) *Mean time spent travelling under 10km/h*

(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 5.8: *PMI results for the six algorithms of §5.1 in the context of a 3×4 grid of intersections under heavier traffic conditions.*

PMI	Mean value						*p*-value	
	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	ANOVA	Levene's Test
1	82.28	71.37	70.94	77.90	66.41	57.97	$<1\times10^{-17}$	7.37×10^{-10}
2	1.875	1.762	1.755	1.834	1.711	1.630	$<1\times10^{-17}$	2.82×10^{-10}
3	2.382	2.272	1.936	3.359	1.637	1.745	$<1\times10^{-17}$	5.83×10^{-13}
4	0.731	0.702	0.599	1.038	0.516	0.575	$<1\times10^{-17}$	1.35×10^{-14}
5	48.44	39.52	38.50	44.64	35.37	28.26	$<1\times10^{-17}$	1.14×10^{-10}
6	0.2574	0.2231	0.2181	0.2359	0.2065	0.1682	$<1\times10^{-17}$	4.48×10^{-10}

TABLE 5.8: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under heavier traffic conditions in a 3×4 grid of intersections. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

In terms of the mean number of stops and the normalised mean number of stops, Gersh went from being the best performing algorithm under lighter traffic conditions, to the overall worst under heavier traffic conditions. LH and Fixed were the superior algorithms in terms of both of these PMIs under heavier traffic conditions which may be seen Figures 5.8(c) and 5.8(d), and Tables 5.11 and 5.12), outperforming Hybrid significantly, which was the next best performing algorithm (achieving corresponding values of 1.936 and 0.599). Fixed was the best performing algorithm over all the PMIs, achieving a mean delay time of 57.97 seconds, 8.44 seconds less than the next shortest mean delay time (see Figure 5.8(a) and Table 5.9).

When these results are compared to those reported by Einhorn [16] for heavier traffic conditions, there are once again differences. In terms of mean delay time and normalised mean delay time, Einhorn [16] found that the O-TSCA was the best performing algorithm, followed by Gersh in both cases. This stands in contrast to the findings reported in Figure 5.8 in which LH is the best performing algorithm and Gersh is the second worst performing algorithm. Einhorn [16] also reported that O-TSCA was once again the most efficient at preventing vehicle stops, followed by Hybrid. This is again inconsistent with what is reported in Figure 5.8.

5.2.3 Result differences

There are a number of possible causes for the differences between the results in Figures 5.7–5.8 and those reported by Einhorn [16], the obvious reason being the use of different simulation models. The literature review of Chapter 2 provided insight as to the expected behaviour of the algorithms of §5.1, which in some cases was different to what was observed in the simulations performed in this dissertation. For instance, it was reported by Einhorn [16] that the I-TSCA switched signals frequently (which was the reason for its success under light traffic conditions). While it does the same in the simulation model of this dissertation, signals are switched so frequently that often only one vehicle makes it through the intersection at a time. It is speculated that perhaps the simulation model of Einhorn [16] contained vehicles with quicker acceleration rates, allowing more vehicles through intersections per time unit. Supporting evidence for this speculation is offered by the results returned by the fixed control scheme employed by Einhorn [16]. In this fixed control scheme, a value of two seconds was implemented for lighter traffic flows for each green phase, implying that two seconds would be sufficient to allow at least one vehicle to make it through the intersection in the simulation model created by Einhorn, while a minimum of 3.33 seconds is necessary for one initially stationary vehicle to make it through the intersection in the model implemented in this dissertation. In the simulation model implemented

p-values of the Games-Howell test: Mean delay						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	7.92×10^{-12}	1.35×10^{-11}	9.50×10^{-9}	1.01×10^{-11}	1.48×10^{-11}
O-TSCA		—	8.12×10^{-1}	1.17×10^{-12}	$<1 \times 10^{-17}$	6.22×10^{-12}
Hybrid			—	8.06×10^{-13}	3.95×10^{-12}	1.27×10^{-11}
Gersh				—	$<1 \times 10^{-17}$	5.70×10^{-14}
LH					—	1.14×10^{-11}
Fixed						—
Mean	82.28	71.37	70.94	77.90	66.41	57.97

TABLE 5.9: *Differences in respect of the mean delay obtained for each of the original self-organising algorithms under heavier traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean delay						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	8.97×10^{-12}	1.38×10^{-11}	1.19×10^{-7}	1.19×10^{-11}	1.46×10^{-11}
O-TSCA		—	3.05×10^{-1}	1.11×10^{-12}	$<1 \times 10^{-17}$	1.13×10^{-11}
Hybrid			—	4.94×10^{-13}	7.23×10^{-12}	1.46×10^{-11}
Gersh				—	$<1 \times 10^{-17}$	2.16×10^{-13}
LH					—	9.78×10^{-12}
Fixed						—
Mean	1.875	1.762	1.755	1.834	1.711	1.630

TABLE 5.10: *Differences in respect of the normalised mean delay obtained for each of the original self-organising algorithms under heavier traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	8.63×10^{-3}	1.33×10^{-11}	$<1 \times 10^{-17}$	1.47×10^{-11}	$<1 \times 10^{-17}$
O-TSCA		—	1.49×10^{-11}	$<1 \times 10^{-17}$	1.38×10^{-11}	3.31×10^{-12}
Hybrid			—	$<1 \times 10^{-17}$	1.72×10^{-11}	4.37×10^{-5}
Gersh				—	$<1 \times 10^{-17}$	1.05×10^{-12}
LH					—	6.46×10^{-2}
Fixed						—
Mean	2.382	2.272	1.936	3.359	1.637	1.745

TABLE 5.11: *Differences in respect of the mean number of stops obtained for each of the original self-organising algorithms under heavier traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	2.40×10^{-2}	1.45×10^{-11}	$<1 \times 10^{-17}$	1.49×10^{-11}	$<1 \times 10^{-17}$
O-TSCA		—	1.46×10^{-11}	$<1 \times 10^{-17}$	1.37×10^{-11}	4.20×10^{-12}
Hybrid			—	$<1 \times 10^{-17}$	2.77×10^{-11}	2.25×10^{-1}
Gersh				—	$<1 \times 10^{-17}$	9.02×10^{-13}
LH					—	1.65×10^{-5}
Fixed						—
Mean	0.731	0.702	0.599	1.038	0.516	0.575

TABLE 5.12: *Differences in respect of the normalised mean number of stops obtained for each of the original self-organising algorithms under heavier traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean time spent travelling under 10km/h						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	1.22×10^{-11}	1.42×10^{-11}	5.81×10^{-9}	6.75×10^{-12}	1.49×10^{-11}
O-TSCA		—	5.89×10^{-3}	2.15×10^{-12}	$<1 \times 10^{-17}$	1.19×10^{-11}
Hybrid			—	4.95×10^{-13}	1.92×10^{-12}	1.39×10^{-11}
Gersh				—	$<1 \times 10^{-17}$	2.51×10^{-14}
LH					—	7.38×10^{-12}
Fixed						—
Mean	48.44	39.52	38.50	44.64	35.37	28.26

TABLE 5.13: *Differences in respect of the mean time spent travelling under 10km/h obtained for each of the original self-organising algorithms under heavier traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean time spent travelling under 10km/h						
Algorithm	I-TSCA	O-TSCA	Hybrid	Gersh	LH	Fixed
I-TSCA	—	3.21×10^{-12}	7.28×10^{-12}	$<1 \times 10^{-17}$	1.44×10^{-11}	1.25×10^{-11}
O-TSCA		—	1.74×10^{-4}	4.40×10^{-8}	$<1 \times 10^{-17}$	1.11×10^{-11}
Hybrid			—	6.30×10^{-12}	3.25×10^{-12}	1.35×10^{-11}
Gersh				—	$<1 \times 10^{-17}$	2.15×10^{-13}
LH					—	9.57×10^{-12}
Fixed						—
Mean	0.2574	0.2231	0.2181	0.2359	0.2065	0.1682

TABLE 5.14: *Differences in respect of the normalised mean time spent travelling under 10km/h obtained for each of the original self-organising algorithms under heavier traffic conditions. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

in this dissertation, acceleration and deceleration rates are taken as the values recommended within the Anylogic simulation software environment [3], which are $1.8\,\mathrm{ms}^{-2}$ and $-4.2\,\mathrm{ms}^{-2}$, respectively. While the acceleration and deceleration rates implemented in the model of Einhorn [16] are not known, it is known for certain that those acceleration rates were substantially larger.

5.3 Improvements of the algorithms of Einhorn

The behaviours exhibited by the three algorithms proposed by Einhorn [16] observed in the simulation model of this dissertation are discussed in this section. This is followed by a number of proposed changes to each of these three algorithms. Initially, algorithmic improvements were not envisaged as within the scope of this dissertation, but as a result of the implementation of the algorithms and by observing many simulation runs, various shortcomings of these algorithms were discovered. While improvements to the algorithms proposed by Einhorn [16] are proposed and tested in this section, no changes are suggested for the fixed-time control strategy, the algorithm by Gershenson and Rosenblueth [22] or the algorithm by Lämmer and Helbing [46].

5.3.1 The improved O-TSCA

While it is stated in [16] that the O-TSCA is free of parameters, it does, in fact, indirectly make use of a parameter. The algorithm operates according to the detected demand and availability along approach and exit lanes, respectively. The effective “sight” of the algorithm extends 275 metres down each roadway connected to the intersection, as this is the maximum distance over which the assumed mode of detection functions. Although the O-TSCA makes use of this maximum distance, it is not necessarily the most suitable distance. It is therefore recommended that this parameter be adjusted according to the road network and current traffic conditions.

One of the characteristics of the O-TSCA is that it does not change signals until every initially detected vehicle along the approach receiving service has successfully travelled through the intersection. While this may be seen as an advantage of the algorithm, it becomes problematic under lighter traffic conditions when there are large distances between consecutive vehicles. Consider, for instance, the example in Figure 5.9. The signal has just turned green for the horizontal direction, with $\Delta = 56$, which is equivalent to eight vehicles (since each vehicle has an effective length of seven meters and one vehicle approaching from the right is not visible in the image). The algorithm will only consider switching signals once all of these eight vehicles have passed through the intersection, since at that point in time, $\theta \geq \Delta$. There will be a point when the five vehicles close to the intersection will all have passed through the intersection and the two vehicles in the horizontal direction furthest from the intersection will still be a considerable distance away, while there will be at least four stationary vehicles in the vertical direction waiting for service. This leads to an ineffective use of green time and an increased total vehicle delay.

There may also be room for improvement of the algorithm in terms of not switching signals if there is a vehicle in very close proximity of the intersection. Since the O-TSCA considers changing signals once the initially detected demand has passed through the intersection, it may happen that as signals are changed, a vehicle from an approach receiving service that is very close to the intersection is forced to stop. The delay time of this vehicle will be the sum of at least two setup times as well as the green time received by opposing directions. On the other hand, if the service were to be extended a couple of seconds, the vehicle would not experience additional delay and the other vehicles will each only experience an extra couple of seconds' delay time. While the technique is incorporated in the IUMSM of the Hybrid algorithm and may alleviate the two problematic situations described above, it is proposed that similar techniques are incorporated directly into the I-TSCA and the O-TSCA.

The minimum and maximum green times allowed by the algorithm were varied in order to ascertain whether there is an advantage to limiting the algorithm to certain green time intervals. A minimum green time of 7 seconds was considered [42]. Due to the longer green times associated with this algorithm, this constraint was not necessary, since even if the green time allocated by the O-TSCA was less than 7 seconds (which is relatively rare), all vehicles initially detected had been served by the time that signals were changed. Maximum green times of 40 and 70 seconds for lighter and heavier traffic conditions, respectively, were selected [42], although it was found that this change yielded very similar results and no restriction was therefore set on the lengths of green times.

The original O-TSCA was, however, altered to accommodate the two problematic situations described above. The demand and availability detection length was taken to be 110 metres (in Figure 5.9 this is the sum of the length of the three-lane approach together with the length of the lane merge), while the proximity of vehicles to the intersection that prevent a signal change was varied from 0 to 20 metres. A distance of ten metres was ultimately chosen for this parameter as it yielded the best results for both light and heavy traffic conditions.

5.3.2 The improved I-TSCA

In contrast to the O-TSCA, the disadvantage of the I-TSCA lies in its switching of signals too often, rather than too infrequently. It is found that the majority of the time the initially required

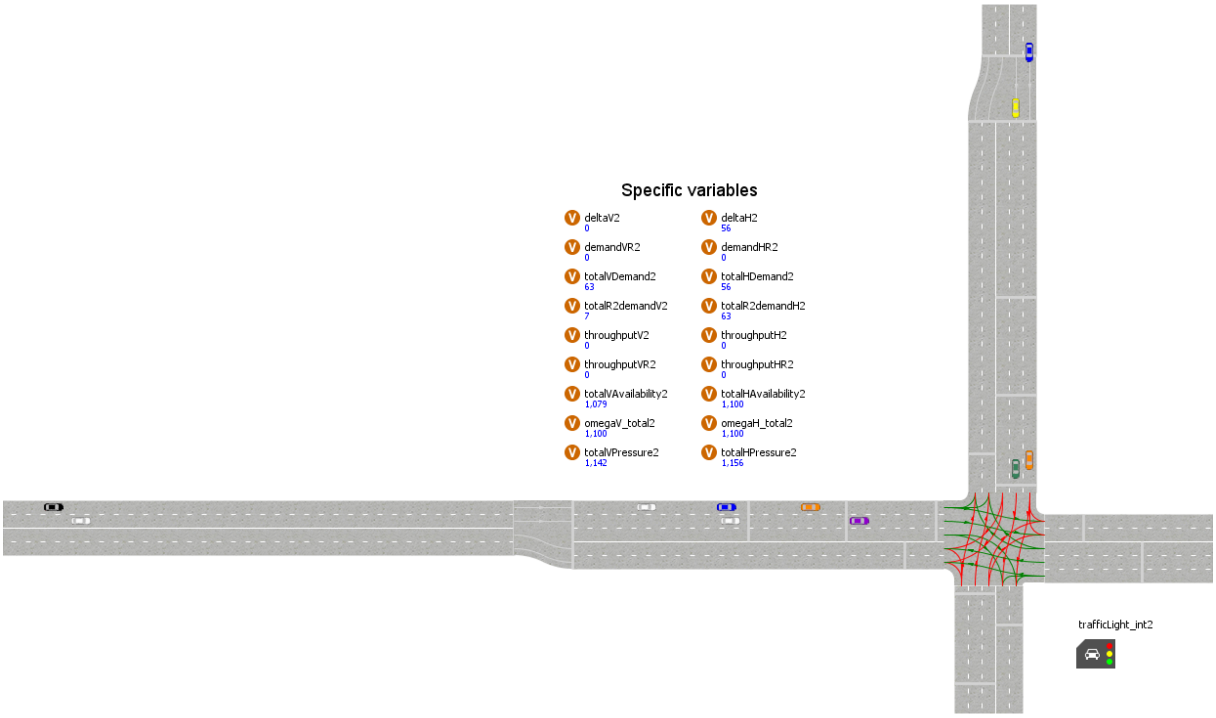


FIGURE 5.9: *An example of ineffective switching of signals by the O-TSCA in the Anylogic simulation model.*

green time (before green time extensions) was 3.33 seconds in the simulation model, as this is the time required for a stationary vehicle at the intersection to travel through it. Once this time has elapsed, the I-TSCA calculates a potential extended green time based on the distance between the closest vehicle and the intersection. If the extension is granted, the length of this extension is the minimum time necessary for a vehicle to cross the intersection. This vehicle is often very near to the intersection and does not, in fact, require an extension because if it were travelling at speed, it would make it through the intersection during the amber phase. This is an example of an unnecessary extension. There are, however, also cases where an extension would be preferable, yet it is not granted. This is caused when delays are not accurately predicted, which is unavoidable considering that these delays are based solely on predictions.

The first change recommended for the I-TSCA is applying a minimum green time of 7 seconds to the algorithm [42] in order to prevent it from switching signals after only 3.33 seconds as it commonly does. Incorporation of a maximum green time is not necessary due to the algorithm's fast signal switching nature.

Preventing signals from switching when there is a vehicle in close proximity of the intersection (as was done for the O-TSCA) was also considered for the I-TSCA. It was, however, found that this yielded no improvement in respect of the performance of the algorithm.

Another recommended alteration of the I-TSCA involves the calculation of the extended green time. The algorithm considers the fastest time taken for a vehicle to cross the intersection and calculates a cost based on that. This means that if a green-time extension is allowed, it will only be long enough to ensure that the single closest vehicle makes it through the intersection, resulting in an increment of the delay time of all vehicles queued in the opposing direction, corresponding to the green time extension that was awarded. Rather than basing this calculation

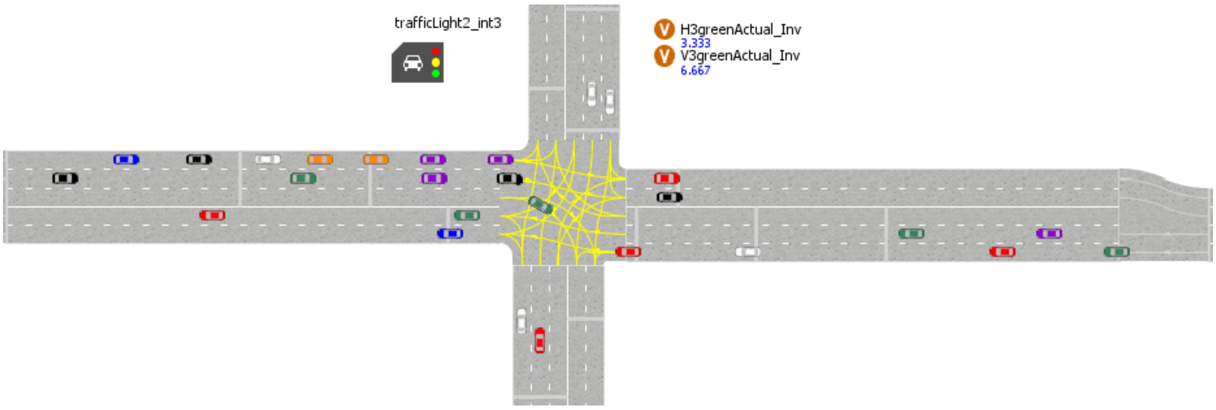


FIGURE 5.10: *An example of ineffective switching of signals by the I-TSCA in the Anylogic simulation model.*

on the vehicle closest to the intersection, it is recommended that vehicles further away are considered as well. In particular, vehicles on the three-lane road section (shown in Figure 5.9) that are furthest from the intersection should also be considered. In this case, required green times will be calculated based on vehicles that are up to 95 metres (the length of the three-lane road) away from the intersection. If a green time extension were to be awarded, in Figure 5.9, for example, it would be the predicted time taken for the white vehicle in the centre lane to reach the intersection (as it will reach the intersection sooner than the white vehicle in the left lane, where both these vehicles are the furthest from the intersection along the three-lane approach). If there is no vehicle along the three-lane approach, then the closest vehicle to the intersection should be considered for the extended green time calculation.

An example of the I-TSCA switching signals prematurely is shown in Figure 5.10. The variables *H3greenActual_Inv* and *V3greenActual_Inv* indicate the most recent horizontal and vertical green times employed at the intersection. The horizontal direction only received 3.33 seconds of green time before switching and it is clear from the figure that there are a number of other vehicles requiring service that are forced to wait for the following cycle(s).

5.3.3 The improved Hybrid algorithm

Hybrid is a combination of the two algorithms considered in §5.3.1 and §5.3.2. Therefore, no additional changes are recommended for this algorithm, other than the indirect changes that will occur as a result of the alterations recommended for the I-TSCA and the O-TSCA.

In §5.2 is was found that Hybrid outperformed or at least matched the performance of both the I-TSCA and the O-TSCA, although its performance is reliant on the performance of these two algorithms. It is therefore expected that if both the I-TSCA and O-TSCA are improved, the Hybrid algorithm should also result in improved performance.

5.4 Algorithmic results after implementing improvements

The simulation results obtained after having implemented the changes proposed in §5.3 to the algorithms by Einhorn [16] are discussed in this section, under both lighter and heavier traffic conditions. These results are represented through the use of box plots and tables which indicate the means of the PMIs, and these results are compared to the results reported in §5.2. The abbreviations I-TSCA(n), O-TSCA(n) and Hybrid(n) refer to the new versions of the algorithms, with the changes implemented as described in §5.3

5.4.1 Results under lighter traffic conditions ($\lambda = 10$ vehicles per minute)

The PMI means returned by the original algorithms, as well as the means returned by their altered counterparts, are shown in Table 5.15. Since the original algorithms are only compared to their altered counterparts, there is no need to conduct *post hoc* tests, as the outcome of the ANOVA will indicate whether the two means values are significantly different. The ANOVA values were smaller than 1×10^{-17} for all six PMIs, implying that all three altered algorithms improved upon the original versions of the algorithms.

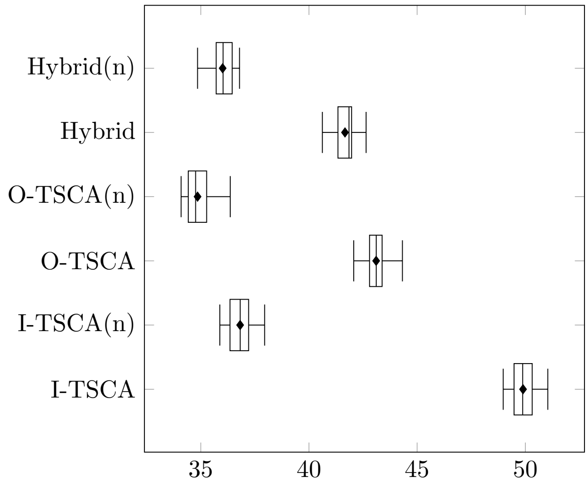
PMI	I-TSCA	I-TSCA(n)	O-TSCA	O-TSCA(n)	Hybrid	Hybrid(n)
1	49.90	36.82	43.11	34.85	41.67	36.01
2	1.534	1.396	1.463	1.376	1.446	1.388
3	0.685	0.277	0.605	0.303	0.413	0.256
4	0.207	0.087	0.184	0.095	0.125	0.082
5	25.56	14.34	21.10	13.40	18.02	13.70
6	0.1665	0.1017	0.1414	0.0964	0.1248	0.0978

TABLE 5.15: *The mean values of the six PMIs under lighter traffic conditions. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

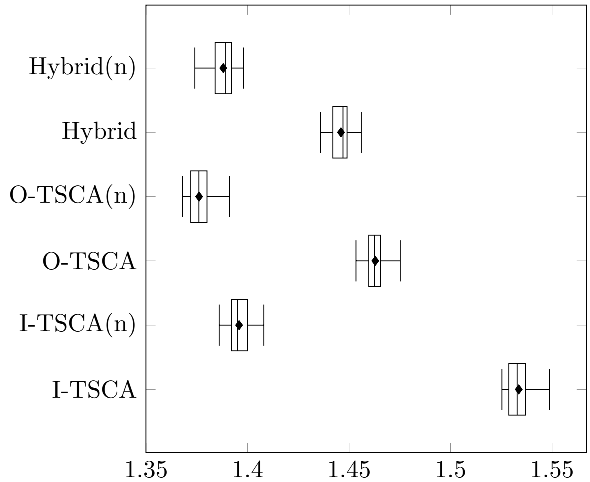
A significant improvement in the performance of each of the three self-organising algorithms is evident in Figure 5.11. The I-TSCA(n) achieved the largest improvement over all of the six PMIs, including a mean delay time reduction of 13.08 seconds (see Figure 5.11(a)) and more than halving the mean number of stops from 0.685 to 0.277 (see Figure 5.11(c)). The mean time spent by vehicles under 10km/h also improved dramatically from 25.56 seconds to 14.34, an improvement of almost 44%.

The I-TSCA(n) was found to switch signals less often than the I-TSCA (due to the minimum green time constraint), yet it still switched signals relatively frequently when necessary, thus maintaining a considerable amount of flexibility. Green time extensions were granted more frequently as a result of the newly required green time calculation in the extension policy, allowing more vehicles to pass through the intersection, on average. It was also found that when a platoon of vehicles was within the three-lane approach of an intersection, the I-TSCA(n) was likely to grant a green time extension if there was not a large number of closely approaching vehicles from the opposing direction.

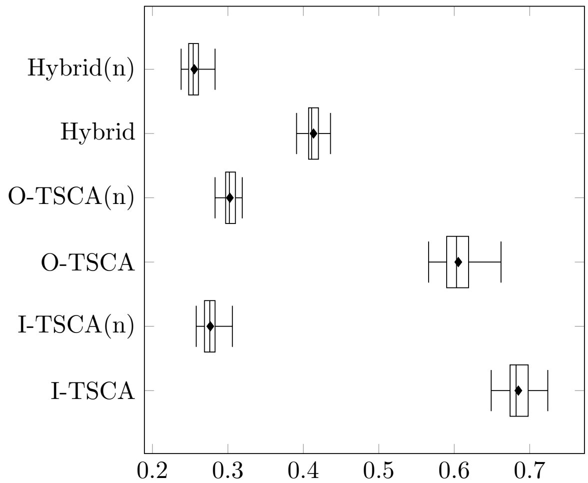
The O-TSCA(n) also achieved a considerable improvement with respect to all six of the PMIs. The mean delay time of the O-TSCA was improved from 43.11 seconds to 34.85 seconds (see Figure 5.11(a)) and its normalised delay from 1.46 to 1.38 (see Figure 5.11(b)). The mean number of stops achieved by the O-TSCA(n) halved and the normalised number of stops decreased by over 48% when compared to the corresponding results of the O-TSCA. The mean time spent under 10km/h improved by 7.70 seconds, from a value of 21.10 to a value of 13.40 (see Figure 5.11(e)).



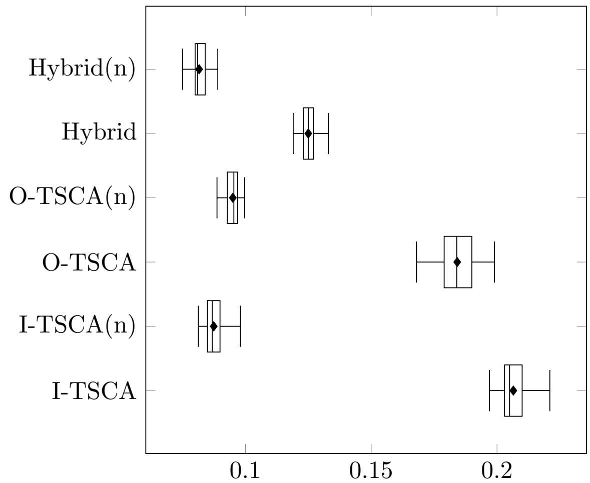
(a) *Mean delay*



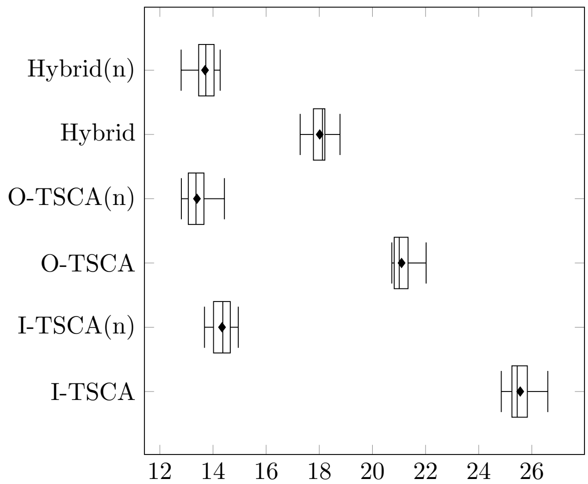
(b) *Normalised mean delay*



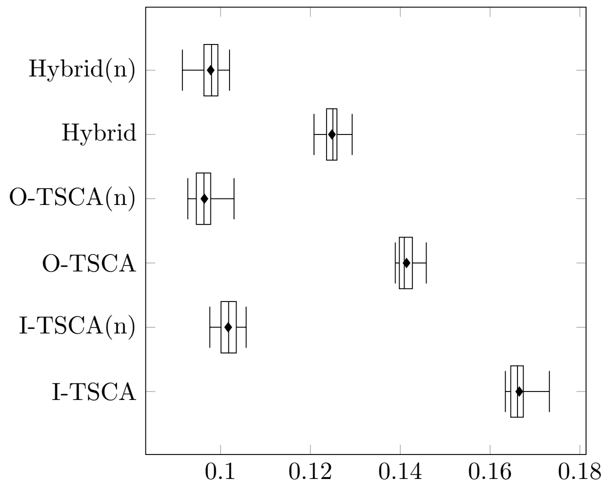
(c) *Mean number of stops*



(d) *Normalised mean number of stops*



(e) *Mean time spent travelling under 10km/h*



(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 5.11: *PMI results for the original and improved self-organising algorithms under lighter traffic conditions.*

Hybrid(n) achieved the smallest improvement comparatively, although it is still a large improvement. This is not surprising, as Hybrid was already the superior algorithm out of the three algorithms proposed by Einhorn [16] before any changes were made. It would therefore be more difficult to improve upon the performance of this algorithm as much as on those of the I-TSCA(n) and the O-TSCA(n). Hybrid(n) achieved a mean delay of 36.01 seconds, a 5.66 second improvement over Hybrid (see Figure 5.11(a)). In terms of mean number of stops and mean time spent under 10km/h, the results of Hybrid were improved upon by 38% and 24%, respectively.

Before any changes were made to Hybrid, it utilised the contrasting elements of the green time allocation by the I-TSCA and the O-TSCA green time allocation to outperform each of the individual algorithms. After the changes to the I-TSCA and O-TSCA were implemented, the green times of the I-TSCA(n) were longer, while the green times of the O-TSCA(n) were shorter than before, resulting in the situation where Hybrid(n) does not follow the same approach any more.

5.4.2 Results under heavier traffic conditions ($\lambda = 20$ vehicles per minute)

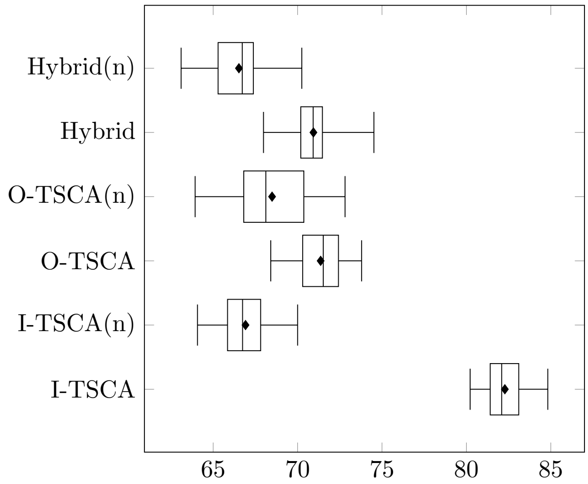
The PMI means returned by the original algorithms, as well as the means returned by their altered counterparts, are shown in Table 5.16. As was the case under lighter traffic conditions, the ANOVA values were smaller than 1×10^{-17} for all six PMIs, implying that all three altered algorithms improved upon the original versions of the algorithm.

PMI	I-TSCA	I-TSCA(n)	O-TSCA	O-TSCA(n)	Hybrid	Hybrid(n)
1	82.28	66.92	71.37	68.49	70.94	66.52
2	1.875	1.716	1.762	1.731	1.755	1.714
3	2.382	1.661	2.272	1.762	1.936	1.762
4	0.731	0.521	0.702	0.547	0.599	0.557
5	48.44	35.38	39.52	36.67	38.50	34.76
6	0.2574	0.2038	0.2231	0.2113	0.2181	0.2015

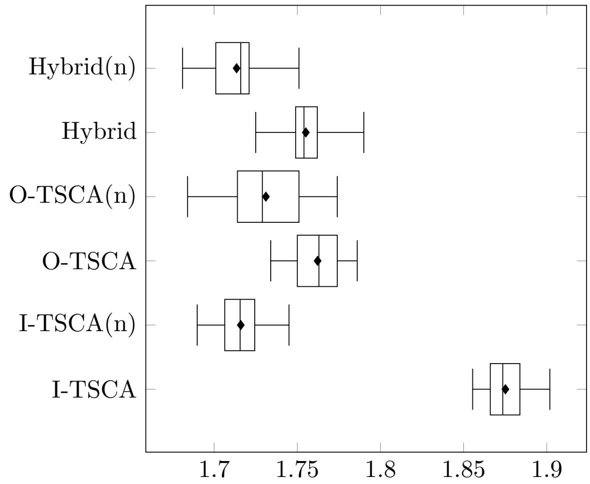
Table 5.16: *The mean values of the six PMIs under heavier traffic conditions. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

As was the case under lighter traffic conditions, all three algorithms proposed by Einhorn [16] were improved with respect to the six PMIs of §4.1.4 under heavier traffic conditions as a result of the algorithmic changes recommended in §5.3, as may be seen in Figure 5.12.

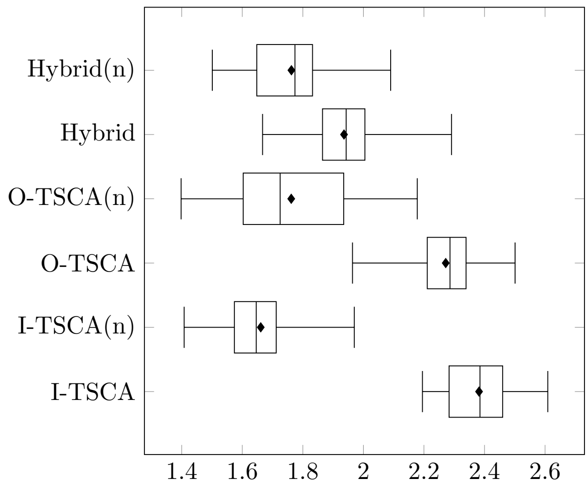
The I-TSCA(n) once again achieved the largest overall improvement, including a 15.36 second reduction in mean delay time from a value of 82.28 seconds to a mean value of 66.92 seconds (see Figure 5.12(a)) as well as a 30% reduction in the mean number of stops made (see Figure 5.12(c)). A 13.06 second improvement of the results of the I-TSCA(n) was observed in terms of the mean time spent under 10km/h over the corresponding results of the I-TSCA, achieving a value of 35.38 seconds (see Figure 5.12(e)) — an indication that the average time a vehicle spends travelling "slowly" is reduced by 64%. The extremely fast switching of the I-TSCA was the principal cause for the poor performance it exhibited before implementation of the changes recommended in §5.3, particularly under heavy traffic conditions. The longer green times resulting from the changes are directly related to the large improvement in the algorithm's performance.



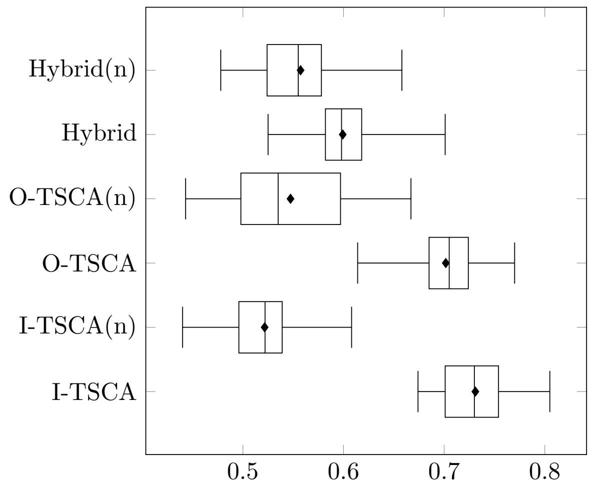
(a) *Mean delay*



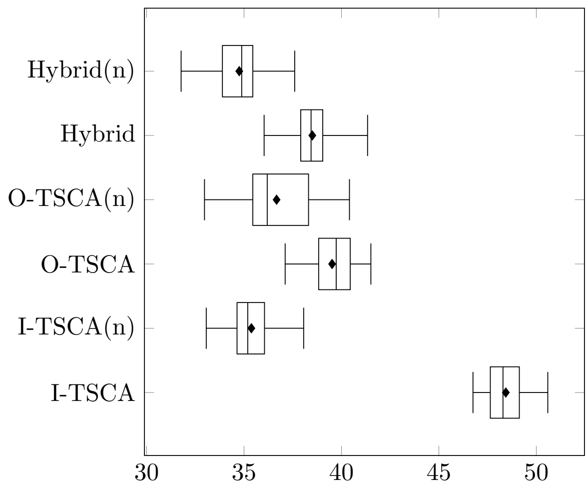
(b) *Normalised mean delay*



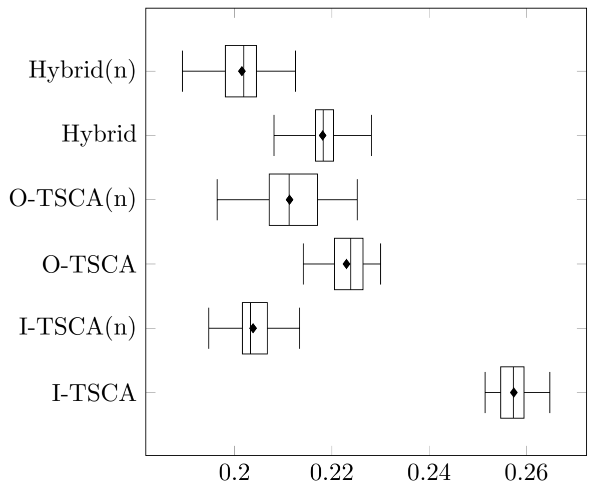
(c) *Mean number of stops*



(d) *Normalised mean number of stops*



(e) *Mean time spent travelling under 10km/h*



(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 5.12: *PMI results for the original and improved self-organising algorithms under heavier traffic conditions.*

The O-TSCA(n) achieved a smaller improvement over the O-TSCA than it did under lighter traffic conditions. The reason for this is that the O-TSCA is, in principle, better suited for heavier traffic conditions (due to its longer green times). The change resulting from considering a smaller section of road, therefore, did not yield such a large performance improvement, as the longer green times are better suited to heavier traffic conditions. The largest improvement of the O-TSCA(n) was in respect of the mean number of stops and normalised mean number of stops, improving in both these PMIs by just over 22% (see Figures 5.12(c) and 5.12(d)) when compared to the corresponding results of the O-TSCA. Improvements in the mean delay and the time spent under 10km/h were of a smaller margin, yet still significant at a 5% significance level, obtaining improvements of 2.87 and 2.85 seconds, respectively (see Figures 5.12(a) and 5.12(e)).

The performance of Hybrid(n) was improved the least out of the three algorithms proposed by Einhorn [16], yet the improvement is still significant at a 5% level of significance. The mean delay and the time spent under 10km/h improved by 4.42 and 3.74 seconds, respectively, while the mean number of stops was reduced by just over 9% in comparison to the original Hybrid.

5.5 Chapter summary

This chapter opened in §5.1 with descriptions of the logic behind a fixed-time control strategy and five of the self-organising algorithms reviewed in Chapter 2, revealing how they were interpreted and implemented in this dissertation. This was followed by a comparison of the results obtained by the six algorithms in §5.2 in terms of the six PMIs of §4.1.4 under both lighter and heavier traffic conditions. Algorithmic changes were proposed for three of the six algorithms in §5.3 and the improvements achieved as a result of implementing these changes were quantified in §5.4. These improvements were significant (by a large margin) at a 5% significance level.

CHAPTER 6

Initial simulation results

Contents

This chapter opens in §6.1 with a brief description of the parameter settings selected for each of the self-organising algorithms and the fixed-time control algorithm. Two scenarios considered by Einhorn [16] are replicated in this chapter. These two scenarios are a four-intersection corridor and a 3×4 grid of intersections, for which simulation results are reported in §6.2 and §6.3, respectively. The chapter closes in §6.4 with a summary of the chapter contents.

6.1 Algorithmic parameter settings

While most of the parameter settings for the six algorithms are mentioned in the descriptions of the individual algorithmic implementations in §5.1, they are also summarised in this section.

The O-TSCA(n) makes use of two parameters, one of which is the length of road that is considered by the algorithm when calculating demand. While a sensitivity analysis of this parameter is recommended so as to find an optimal parameter value, it is chosen to be 95 metres in this dissertation as this value was found to result in a dramatic improvement over the 275 metre value considered by Einhorn [16]. Another parameter accommodates the inclusion of a mechanism (similar to that employed by the IUMSM) which prevents signals from switching while there is a vehicle within a specified distance from the intersection. A range of values were considered for this parameter and it was found that the parameter was most effective when set to 10 metres in a grid or corridor network, for both lighter and heavier traffic conditions.

The I-TSCA(n) makes use of a similar parameter, which is the distance that the algorithm considers when making the decision to extend or terminate a green signal. Einhorn [16] suggested that the distance of the nearest vehicle be considered. It was, however, found that selecting larger distances for this purpose improved the performance of the algorithm considerably. This parameter takes the value of the distance of the vehicle furthest from the intersection along the three-lane approach (which is 80 meters in total). If there is no vehicle on this road section, the distance between the closest vehicle and the intersection is chosen. A property of the I-TSCA(n) is that it selects the minimum required green time across all approach lanes, which is often only 3.33 seconds (equivalent to the time taken for one stationary vehicle to travel through the intersection). In order to overcome this shortcoming, a minimum green time of 7 seconds is enforced.

Since Hybrid(n) is a combination of the I-TSCA(n) and O-TSCA(n), it makes use of the above-mentioned four parameters.

The Gersh algorithm makes use of six parameters that were described in §2.1.1. The values recommended by Gershenson and Rosenblueth in [22] are used in this dissertation. The distances $d = 50$ metres, $r = 25$ metres and $e = 10$ metres are used in the simulation model. The counter n is set to 53.33, while the number of vehicles s is set to the recommended value of 2. The only recommended value that is not used is the minimum green time of 3.33 seconds, since it was found to be too short, and thus a minimum green time of 7 seconds is used instead.

The LH algorithm employs only two parameters which are both related to the stabilisation strategy implemented when queues become unstable. The values of these parameters are taken as in the example in [47], where Z is set to 90 seconds and Z^{max} is set to 120 seconds.

The cycle length utilised by Fixed is 30 seconds for lighter traffic flow densities and 60 seconds for heavier traffic flow densities. The resulting green times under lighter traffic conditions are therefore 10 seconds long, while 25-second green times are employed under heavier traffic conditions.

The two road network topologies considered in the study by Einhorn [16] were a four-intersection road corridor and a 3×4 grid of intersections. For the sake of completeness, the improved self-organising algorithms of §5.3 are compared to one another as well as to the algorithm by Gershenson and Rosenblueth [22] and the algorithm by Lämmer and Helbing [46] in these two scenarios, under both lighter and heavier traffic conditions, but in the new traffic simulation model developed in this dissertation. The corresponding results obtained by Einhorn [16] are also mentioned briefly. The results are presented in the form of box plots and *post hoc* tables indicating 5% significance intervals for the PMIs achieved by each algorithm.

6.2 A four-intersection road corridor

A corridor of four equally spaced intersections is the road network topology considered in this section, as depicted in Figure 6.1. It was found by Einhorn [16] that Hybrid was the overall best performing algorithm in this network topology for vehicle arrivals according to Poisson distributions with rates of $\lambda = 10$ vehicles per minute and $\lambda = 20$ vehicles per minute.

Simulation results for lighter traffic conditions ($\lambda = 10$ vehicles per minute)

The ANOVA column in Table 6.1 indicates that for all six PMIs there are statistical differences between the means returned by the algorithms at a 5% level of significance. Furthermore, the outcome of the Levene test revealed that the algorithmic variances of the mean samples are not statistically distinguishable for any of the six PMIs at a 5% level of significance. Therefore, the Fisher LSD test is employed in all cases to determine between which pairs of algorithmic outputs differences between the PMI means are discernible.

Hybrid(n) and the I-TSCA(n) achieved the best results in terms of mean delay time and normalised mean delay time (shown in Figures 6.2(a) and 6.2(b) as well as in Tables 6.2 and 6.3),

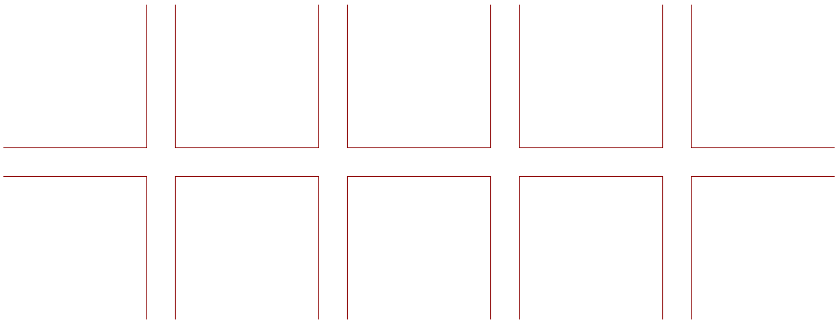


FIGURE 6.1: *A corridor of four homogeneous intersections.*

	Mean value						p-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	ANOVA	Levene's Test
1	18.04	18.55	17.95	18.30	20.29	19.34	$<1 \times 10^{-17}$	4.30×10^{-1}
2	1.327	1.338	1.327	1.332	1.363	1.358	$<1 \times 10^{-17}$	7.37×10^{-1}
3	0.125	0.106	0.118	0.108	0.213	0.178	$<1 \times 10^{-17}$	5.16×10^{-1}
4	0.074	0.066	0.072	0.068	0.125	0.102	$<1 \times 10^{-17}$	5.75×10^{-1}
5	7.28	7.54	7.21	7.45	9.22	8.39	$<1 \times 10^{-17}$	7.39×10^{-1}
6	0.0879	0.0929	0.0885	0.0902	0.1055	0.1040	$<1 \times 10^{-17}$	3.23×10^{-1}

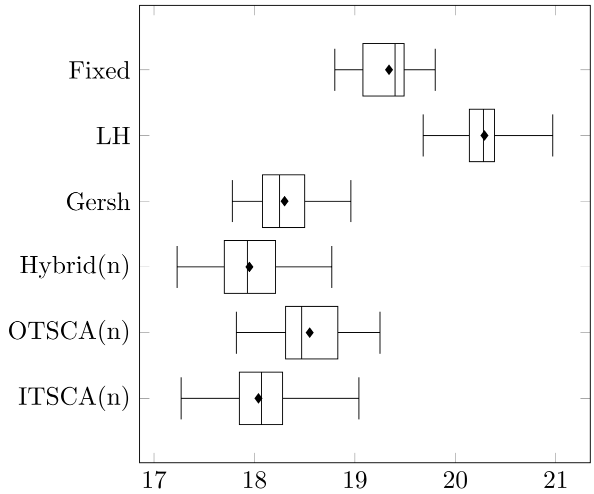
TABLE 6.1: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under lighter traffic conditions in a four-intersection corridor. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

statistically outperforming all other algorithms, but not differing significantly from one another. The next favourable PMI result in terms of both these measures was obtained by Gersh which achieved a mean delay of 18.3 seconds and a normalised mean delay of 1.332, indicating that vehicles under control of Gersh spend an additional 33.2% of their time in the road network as a result of delays. LH was outperformed by all algorithms in terms of both these measures.

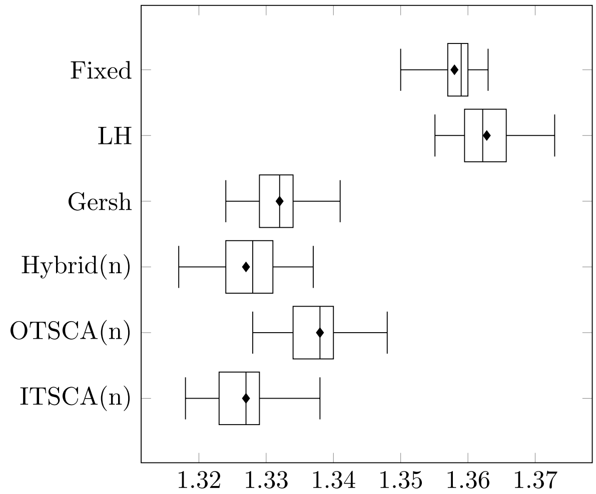
Gersh and the O-TSCA(n) were the most effective in preventing vehicle stops, statistically outperforming all other algorithms in terms of mean number of stops and normalised mean number of stops, while not differing significantly from one another. The I-TSCA(n) and Hybrid(n) did not perform statistically differently from one another in terms of these two measures, as is evident in Tables 6.4 and 6.5. They did, however, significantly outperform LH and Fixed, achieving PMI values almost half of those achieved by LH, which obtained a mean number of stops value of 0.213 and a normalised mean number of stops value of 0.125.

Hybrid(n) and the I-TSCA(n) achieved the smallest values for mean time vehicles spend travelling under 10km/h as well as the normalised mean time vehicles spend travelling under 10km/h (see Figures 6.2(e) and 6.2(f) as well as Tables 6.6 and 6.7), while once again not differing significantly from one another. Hybrid(n) outperformed Gersh by 3.22%, the O-TSCA(n) by 4.38%, Fixed by 14.1% and LH by 21.8% in respect of mean time travelling under 10km/h.

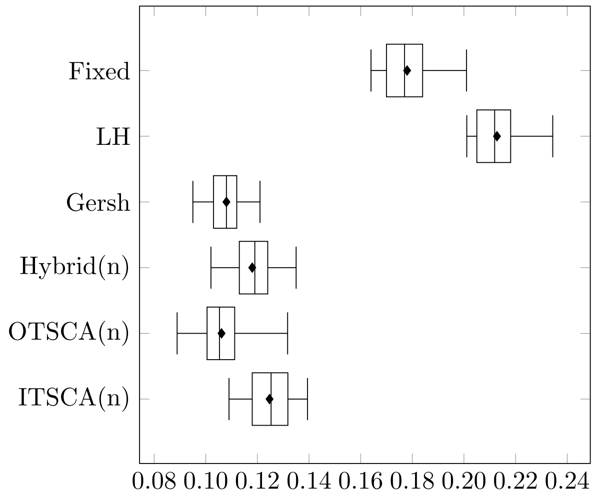
The results obtained by Einhorn [16] indicated that Hybrid was the best performing algorithm in respect of both mean delay time and normalised mean delay time, which corresponds to what was found in respect of Hybrid(n) in this study, although the I-TSCA did not perform as well as the I-TSCA(n) in terms of these measures, as was expected. Einhorn [16] reported that the



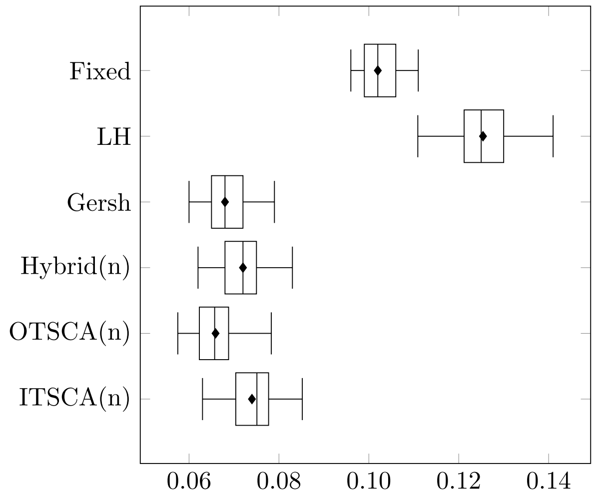
(a) *Mean delay*



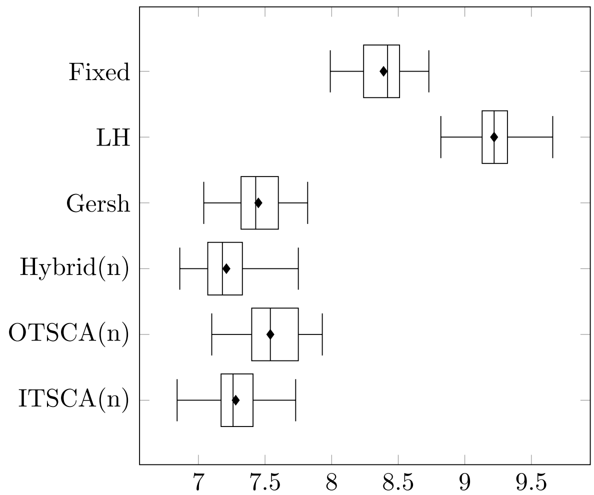
(b) *Normalised mean delay*



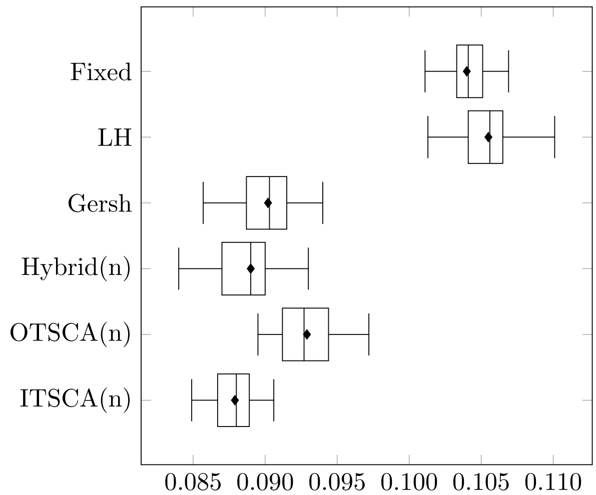
(c) *Mean number of stops*



(d) *Normalised mean number of stops*



(e) *Mean time spent travelling under 10km/h*



(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 6.2: *PMI results for the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor under lighter traffic conditions.*

p-values of the Fisher LSD test: Mean delay						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	4.85×10^{-8}	2.76×10^{-1}	3.84×10^{-3}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
O-TSCA(n)		—	1.61×10^{-10}	6.09×10^{-3}	$<1 \times 10^{-17}$	4.44×10^{-16}
Hybrid(n)			—	8.57×10^{-5}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$
Fixed						—
Mean	18.04	18.55	17.95	18.30	20.29	19.34

Table 6.2: *Differences in respect of the mean delay obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Normalised mean delay						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	$<1 \times 10^{-17}$	5.35×10^{-1}	1.18×10^{-5}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
O-TSCA(n)		—	1.11×10^{-16}	3.17×10^{-7}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Hybrid(n)			—	1.42×10^{-4}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
LH					—	5.99×10^{-5}
Fixed						—
Mean	1.327	1.338	1.327	1.332	1.363	1.358

Table 6.3: *Differences in respect of the normalised mean delay obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Mean number of stops						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	2.11×10^{-15}	2.50×10^{-3}	2.04×10^{-13}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
O-TSCA(n)		—	6.21×10^{-8}	4.50×10^{-1}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Hybrid(n)			—	2.19×10^{-6}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$
Fixed						—
Mean	0.125	0.106	0.118	0.108	0.213	0.178

Table 6.4: *Differences in respect of the mean number of stops obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Normalised mean number of stops						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	1.81×10^{-8}	1.04×10^{-1}	7.70×10^{-5}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
O-TSCA(n)		—	3.17×10^{-5}	6.53×10^{-2}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Hybrid(n)			—	1.66×10^{-2}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$
Fixed						—
Mean	0.074	0.066	0.072	0.068	0.125	0.102

Table 6.5: *Differences in respect of the normalised mean number of stops obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Mean time spent travelling under 10km/h						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	5.21×10^{-7}	2.12×10^{-1}	7.12×10^{-4}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
O-TSCA(n)		—	9.74×10^{-10}	7.89×10^{-2}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Hybrid(n)			—	5.27×10^{-6}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$
Fixed						—
Mean	7.28	7.54	7.21	7.45	9.22	8.39

TABLE 6.6: *Differences in respect of the mean time spent travelling under 10km/h obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Normalised mean time spent travelling under 10km/h						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	$<1 \times 10^{-17}$	1.91×10^{-1}	2.36×10^{-6}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
O-TSCA(n)		—	1.11×10^{-16}	5.22×10^{-8}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Hybrid(n)			—	4.61×10^{-4}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
LH					—	1.10×10^{-3}
Fixed						—
Mean	0.0879	0.0929	0.0885	0.0902	0.1055	0.1040

TABLE 6.7: *Differences in respect of the normalised mean time spent travelling under 10km/h obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

O-TSCA was the most effective in reducing the mean number of vehicle stops while LH was most effective in reducing the normalised mean number of stops. In this study, however, while the O-TSCA(n) and Gersh were found to be the superior algorithms in respect of both these PMI values, LH did not statistically outperform any algorithms in respect of normalised mean number of stops.

There is no clear "best" algorithm apparent for vehicles travelling through a four-intersection corridor under lighter traffic conditions. The I-TSCA(n) and Hybrid(n) were, on average, superior in terms of minimising the delays experienced by vehicles as seen from their delay measures as well as their measures involving the mean time vehicles spend travelling under 10km/h. The O-TSCA(n) and Gersh, on the other hand, were more successful in minimising the number of times that vehicles were required to stop at intersections. It is clear that LH is not suitable for controlling intersections during low density traffic flows, as it achieved the worst PMI values over all six of the measures.

Simulation results for heavier traffic conditions ($\lambda = 20$ vehicles per minute)

As was the case under lighter traffic conditions, the ANOVA column in Table 6.8 indicates that for all six PMIs there are statistical differences between the means of the algorithms at a 5% level of significance. Furthermore, the outcome of the Levene test revealed that the algorithmic variances of the mean samples are significantly different for all six of the PMIs. The Games-Howell *post hoc* test is therefore employed to determine between which pairs of algorithmic outputs differences between the PMI means are discernible.

	Mean value						*p*-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	ANOVA	Levene's Test
1	33.58	33.06	33.00	36.22	33.37	31.27	$<1 \times 10^{-17}$	3.81×10^{-2}
2	1.601	1.598	1.594	1.688	1.609	1.599	$<1 \times 10^{-17}$	1.25×10^{-3}
3	0.864	0.768	0.864	1.389	0.850	0.957	$<1 \times 10^{-17}$	6.68×10^{-5}
4	0.504	0.458	0.510	0.938	0.518	0.632	$<1 \times 10^{-17}$	1.50×10^{-6}
5	18.09	17.56	17.47	20.42	18.01	16.46	$<1 \times 10^{-17}$	4.15×10^{-3}
6	0.1739	0.1765	0.1707	0.1986	0.1796	0.1735	$<1 \times 10^{-17}$	6.68×10^{-3}

TABLE 6.8: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under heavier traffic conditions in a four-intersection corridor. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

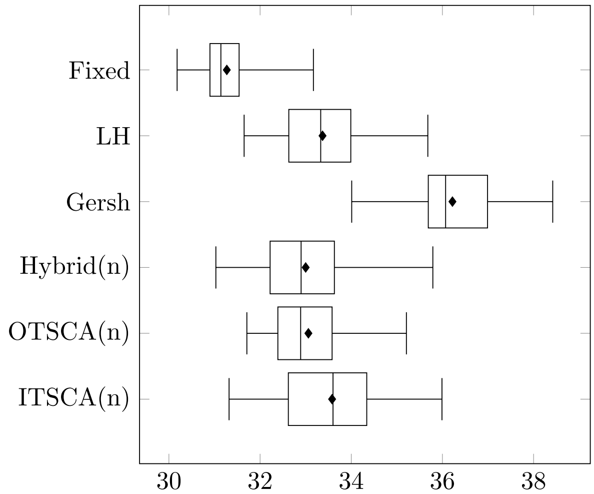
Fixed outperformed all the other algorithms in terms of mean delay time and normalised mean delay time, achieving values of 31.27 seconds and 1.599, respectively. The O-TSCA(n) did not achieve significantly different results than did I-TSCA(n), Hybrid(n) or LH in respect of either of these measures. Gersh achieved the largest values for both mean delay time and normalised mean delay time, obtaining values of 36.22 seconds and 1.688, respectively (see Figures 6.3(a) as well as in 6.3(b) and Tables 6.9 and 6.10).

The O-TSCA(n) achieved the smallest mean number of stops (0.768) as well as the smallest normalised mean number of stops (0.458), significantly outperforming all other algorithms with respect to these PMIs. The O-TSCA(n) outperformed Hybrid(n), LH, the I-TSCA(n), Fixed and Gersh by 8.42%, 9.73%, 11.11%, 19.7% and 44.74%, respectively, in terms of mean number of stops. Hybrid(n), LH and the I-TSCA(n) did not significantly differ from one another in either of these PMIs, although they did all outperform Fixed and Gersh by significant margins. This can be seen in Figures 6.3(c) and 6.3(d) as well as in Tables 6.11 and 6.12.

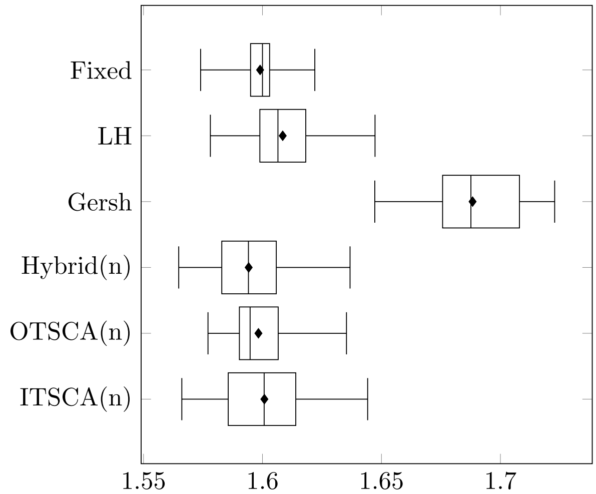
Fixed outperformed all other algorithms in respect of the mean time vehicles spend travelling under 10km/h, although this is not the case for normalised mean time vehicles spend travelling under 10km/h, where the performance of Fixed is not significantly distinguishable from those of I-TSCA(n) or Hybrid(n) at a 5% level of significance. Gersh once again obtained the largest value of this PMI, achieving a value of 20.42 seconds, 18% larger than the smallest mean obtained by Hybrid(n), which obtained a value of 17.25 seconds. In terms of normalised mean time spent by vehicles travelling under 10km/h, Hybrid(n) outperformed all other algorithms besides I-TSCA(n) and Fixed. The performance of Hybrid(n) did not differ significantly from those of the latter two algorithms in respect of the normalised mean time spent by vehicles travelling under 10km/h.

According to the results obtained by Einhorn [16], Gersh achieved the smallest mean delay and normalised mean delay time. In this study, however, it was Fixed that performed the best over all in respect of these two PMIs. With respect to mean number of stops and normalised mean number of stops, however, similar results to those reported by Einhorn [16] were obtained, with the O-TSCA(n) achieving the smallest values for both PMIs, followed closely by Hybrid(n).

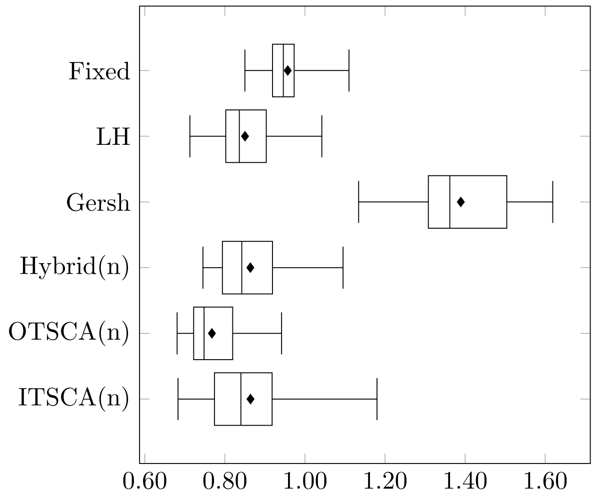
Gersh was statistically the worst performing algorithm in the context of a four-intersection road corridor under heavy traffic conditions, as may clearly be seen in Figure 6.3. The O-TSCA(n) and Fixed are the overall best performing algorithms in respect of the six PMIs, while Hybrid(n) was the second-best performing algorithm overall.



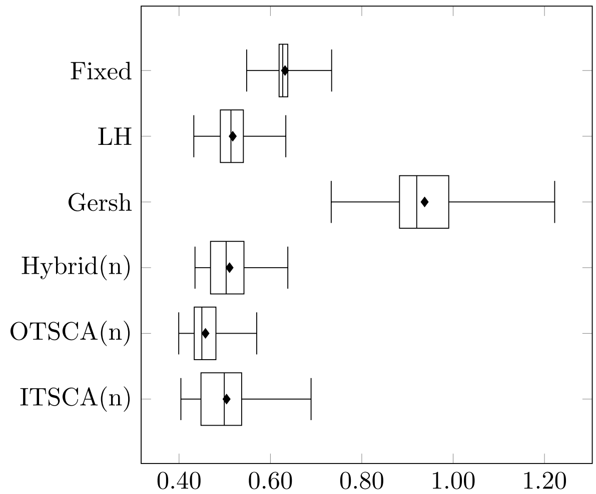
(a) *Mean delay*



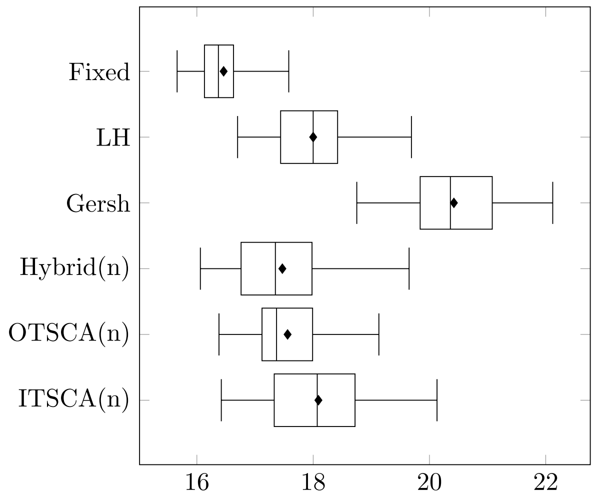
(b) *Normalised mean delay*



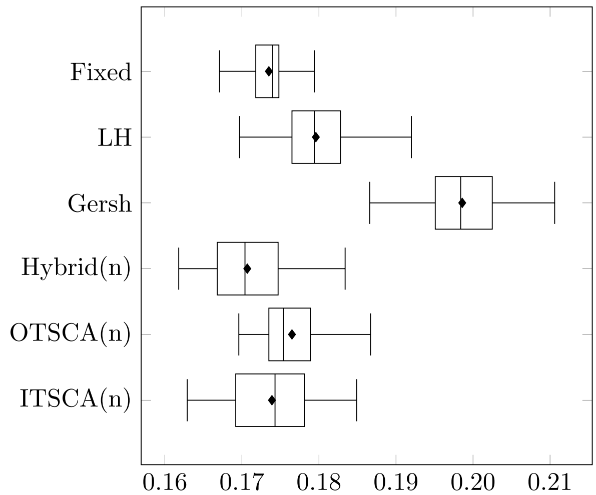
(c) *Mean number of stops*



(d) *Normalised mean number of stops*



(e) *Mean time spent travelling under 10km/h*



(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 6.3: *PMI results for the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor under heavier traffic conditions.*

p-values of the Games-Howell test: Mean delay						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	4.26×10^{-1}	3.70×10^{-1}	3.49×10^{-11}	9.74×10^{-1}	1.74×10^{-10}
O-TSCA(n)		—	9.99×10^{-1}	6.90×10^{-12}	7.68×10^{-1}	3.00×10^{-11}
Hybrid(n)			—	1.49×10^{-11}	6.89×10^{-1}	6.78×10^{-9}
Gersh				—	1.15×10^{-11}	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$
Fixed						—
Mean	33.58	33.06	33.00	36.22	33.37	31.27

TABLE 6.9: *Differences in respect of the mean delay obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean delay						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	9.92×10^{-1}	7.33×10^{-1}	1.36×10^{-11}	5.61×10^{-1}	9.99×10^{-1}
O-TSCA(n)		—	8.99×10^{-1}	$<1 \times 10^{-17}$	8.51×10^{-2}	9.99×10^{-1}
Hybrid(n)			—	2.20×10^{-12}	1.38×10^{-2}	6.88×10^{-1}
Gersh				—	$<1 \times 10^{-17}$	8.33×10^{-13}
LH					—	9.75×10^{-2}
Fixed						—
Mean	1.601	1.598	1.594	1.688	1.609	1.599

TABLE 6.10: *Differences in respect of the normalised mean delay obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	5.22×10^{-3}	9.99×10^{-1}	1.35×10^{-11}	9.94×10^{-1}	7.17×10^{-3}
O-TSCA(n)		—	3.02×10^{-4}	1.13×10^{-12}	3.30×10^{-4}	1.37×10^{-11}
Hybrid(n)			—	$<1 \times 10^{-17}$	9.89×10^{-1}	3.37×10^{-4}
Gersh				—	1.77×10^{-13}	8.22×10^{-13}
LH					—	$<1 \times 10^{-17}$
Fixed						—
Mean	0.864	0.768	0.864	1.389	0.850	0.957

TABLE 6.11: *Differences in respect of the mean number of stops obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	2.08×10^{-2}	9.98×10^{-1}	$<1 \times 10^{-17}$	9.43×10^{-1}	9.08×10^{-11}
O-TSCA(n)		—	7.14×10^{-4}	$<1 \times 10^{-17}$	1.73×10^{-5}	1.49×10^{-11}
Hybrid(n)			—	1.04×10^{-12}	9.94×10^{-1}	$<1 \times 10^{-17}$
Gersh				—	1.00×10^{-13}	$<1 \times 10^{-17}$
LH					—	6.66×10^{-12}
Fixed						—
Mean	0.504	0.458	0.510	0.938	0.518	0.632

TABLE 6.12: *Differences in respect of the normalised mean number of stops obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean time spent travelling under 10km/h						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	1.71×10^{-1}	1.14×10^{-1}	1.15×10^{-11}	9.99×10^{-1}	7.27×10^{-9}
O-TSCA(n)		—	9.98×10^{-1}	3.17×10^{-12}	1.30×10^{-1}	1.90×10^{-8}
Hybrid(n)			—	1.46×10^{-11}	9.03×10^{-2}	6.83×10^{-6}
Gersh				—	5.79×10^{-12}	3.14×10^{-13}
LH					—	$<1 \times 10^{-17}$
Fixed						—
Mean	18.09	17.56	17.47	20.42	18.01	16.46

TABLE 6.13: *Differences in respect of the mean time spent travelling under 10km/h obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean time spent travelling under 10km/h						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	4.16×10^{-1}	2.22×10^{-1}	1.48×10^{-11}	2.72×10^{-3}	9.99×10^{-1}
O-TSCA(n)		—	2.20×10^{-4}	$<1 \times 10^{-17}$	1.17×10^{-1}	3.31×10^{-2}
Hybrid(n)			—	1.21×10^{-11}	1.38×10^{-7}	1.07×10^{-1}
Gersh				—	1.21×10^{-11}	1.10×10^{-12}
LH					—	1.49×10^{-5}
Fixed						—
Mean	0.1739	0.1765	0.1707	0.1986	0.1796	0.1735

TABLE 6.14: *Differences in respect of the normalised mean time spent travelling under 10km/h obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

6.3 A 3×4 grid of intersections

In this section, a fixed-time strategy and five self-organising algorithms are compared to one another within the context of a 3×4 grid of equally spaced intersections, as shown in Figure 6.4. While a similar comparison took place in §5.2 and §5.4, the comparison in §5.2 involved the original algorithms of Einhorn [16], while in §5.4 only the results of the improved versions of the algorithms by Einhorn were compared, excluding Gersh and LH. The improved algorithms I-TSCA(n), O-TSCA(n) and Hybrid(n) are compared with one another as well as with Gersh and LH in this section. In the similar experiment conducted by Einhorn [16], it was found that the O-TSCA(n) was the best performing algorithm under both lighter and heavier traffic conditions.

Simulation results obtained under lighter traffic conditions ($\lambda = 10$ vehicles/min)

The ANOVA column in Table 6.15 indicates that for all six PMIs there are statistical differences between the PMI means returned by the algorithms at a 5% level of significance. Furthermore, the results of the Levene tests revealed that the variances of the mean samples are statistically distinguishable for PMI 1 and PMI 2 at a 5% level of significance, while there is no such significant difference between the variances of means for PMI 3, PMI 4, PMI 5, and PMI 6. Therefore, the Games-Howell *post hoc* test was performed to determine between which pairs of algorithmic outputs differences between the PMI means are discernible in respect of PMI 1 and PMI 2, while the Fisher LSD *post hoc* test was employed for this purpose in respect of the other four PMIs.

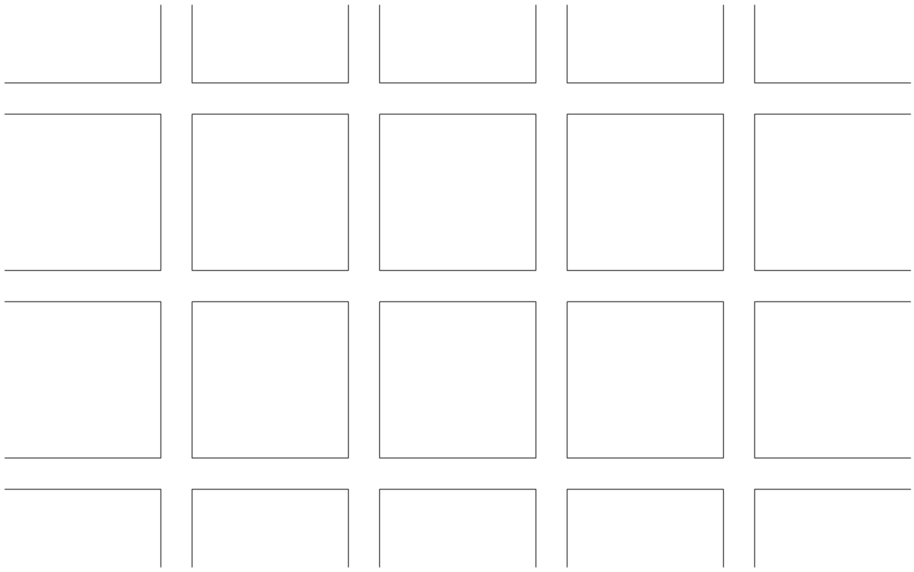


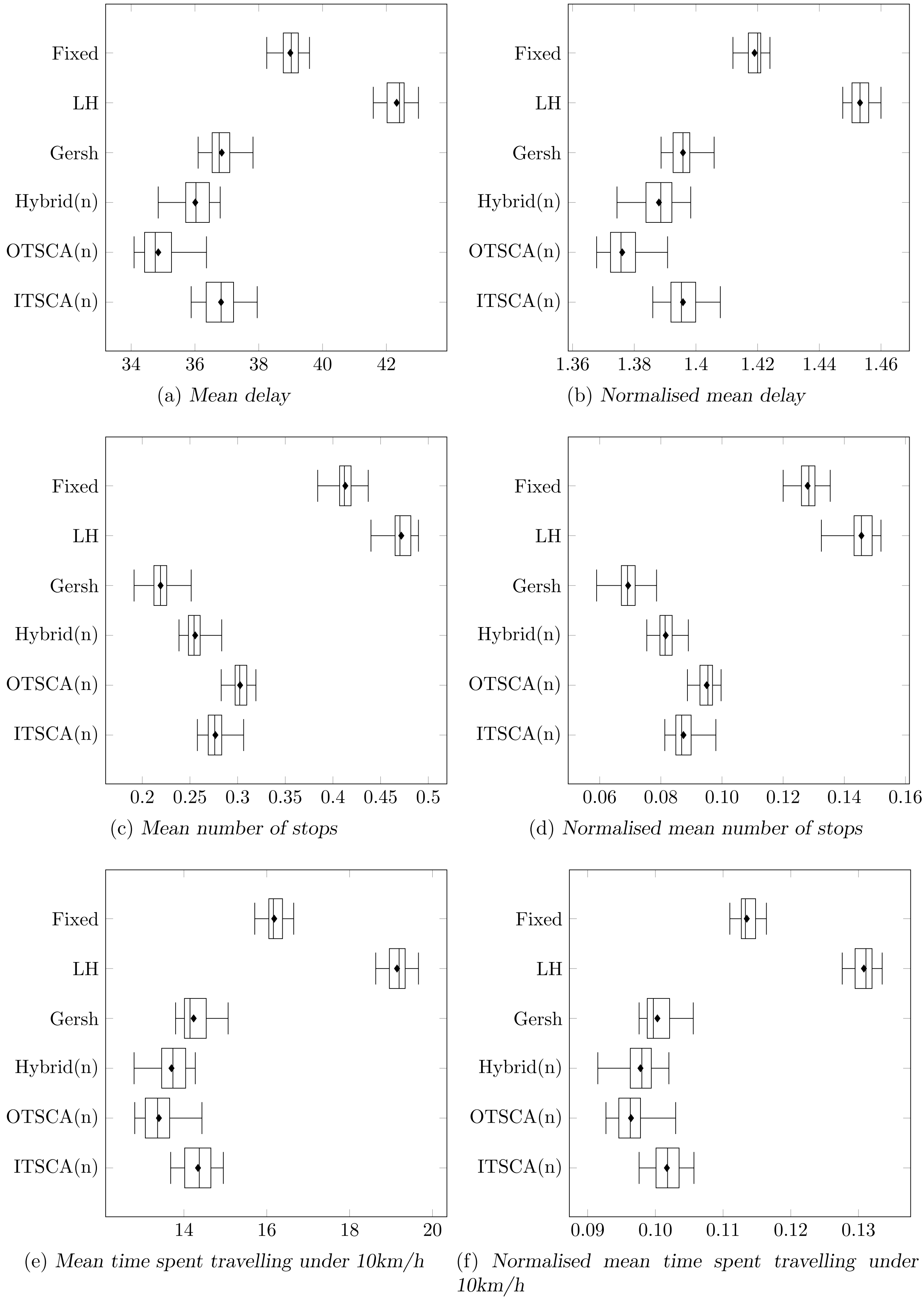
FIGURE 6.4: *A 3×4 regular grid of 12 intersections.*

	Mean value						p-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	ANOVA	Levene's Test
1	36.82	34.85	36.01	36.84	42.32	38.99	$<1 \times 10^{-17}$	2.38×10^{-2}
2	1.396	1.376	1.388	1.396	1.453	1.419	$<1 \times 10^{-17}$	7.33×10^{-3}
3	0.277	0.303	0.256	0.219	0.472	0.413	$<1 \times 10^{-17}$	8.19×10^{-1}
4	0.087	0.095	0.082	0.069	0.146	0.128	$<1 \times 10^{-17}$	8.54×10^{-1}
5	14.34	13.40	13.70	14.24	19.15	16.18	$<1 \times 10^{-17}$	2.21×10^{-1}
6	0.1017	0.0964	0.0978	0.1003	0.1308	0.1135	$<1 \times 10^{-17}$	5.98×10^{-2}

TABLE 6.15: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under lighter traffic conditions in a 3×4 grid of intersections. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

The O-TSCA(n) significantly outperformed the other five algorithms in terms of mean delay and normalised mean delay, obtaining values of 34.85 seconds and 1.376, respectively, as shown in Figures 6.5(a) and 6.5(b) as well as in Tables 6.16 and 6.17. The next best performing algorithm in respect of both these PMIs was Hybrid(n) which obtained corresponding values of 36.01 seconds and 1.388, followed by the I-TSCA(n) and Gersh which did not differ significantly from one another in terms of these PMIs. LH achieved the largest mean delay time of 42.32 seconds and a normalised mean delay time of 1.453, indicating that, on average, vehicles will spent an additional 45.3% of time in the road network as a result of delay, if controlled by LH.

Interestingly, the O-TSCA(n) was less effective in the prevention of vehicle stops than the other algorithms, with the exception of Fixed and LH, as may be seen in Figures 6.5(c) and 6.5(d), as well as in Tables 6.18 and 6.19. Gersh is the best performing algorithm in terms of mean number of stops and normalised mean number of stops made by vehicles, achieving values of 0.219 and 0.069 respectively, which amounts to 14.45% and 15.85% reductions with respect to Hybrid(n) — the next best performing algorithm. LH once again achieved the largest respective PMI values of 0.471 and 0.146, which are respectively 14.0% and 14.1% larger than the corresponding values achieved by Fixed, the second-worst performing algorithm in this respect.



(a) *Mean delay*

(b) *Normalised mean delay*

(c) *Mean number of stops*

(d) *Normalised mean number of stops*

(e) *Mean time spent travelling under 10km/h*

(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 6.5: *PMI results for the five self-organising algorithms and the fixed-time control strategy in the context of a regular 3 × 4 grid of intersections under lighter traffic conditions.*

p-values of the Games-Howell test: Mean delay						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	1.45×10^{-11}	1.18×10^{-5}	9.99×10^{-1}	0.00×10^{-17}	0.00×10^{-17}
O-TSCA(n)		—	6.59×10^{-10}	8.84×10^{-13}	0.00×10^{-17}	0.00×10^{-17}
Hybrid(n)			—	3.03×10^{-7}	0.00×10^{-17}	0.00×10^{-17}
Gersh				—	1.48×10^{-12}	7.50×10^{-12}
LH					—	1.37×10^{-11}
Fixed						—
Mean	36.82	34.85	36.01	36.84	42.32	38.99

TABLE 6.16: *Differences in respect of the mean delay obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean delay						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	1.34×10^{-11}	5.14×10^{-5}	9.99×10^{-1}	4.80×10^{-13}	1.68×10^{-13}
O-TSCA(n)		—	6.22×10^{-10}	6.58×10^{-12}	0.00×10^{-17}	0.00×10^{-17}
Hybrid(n)			—	6.98×10^{-6}	6.00×10^{-13}	3.34×10^{-13}
Gersh				—	0.00×10^{-17}	0.00×10^{-17}
LH					—	1.48×10^{-11}
Fixed						—
Mean	1.396	1.376	1.388	1.396	1.453	1.419

TABLE 6.17: *Differences in respect of the normalised mean delay obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Mean number of stops						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	0.00×10^{-17}	6.32×10^{-13}	0.00×10^{-17}	0.00×10^{-17}	0.00×10^{-17}
O-TSCA(n)		—	0.00×10^{-17}	0.00×10^{-17}	0.00×10^{-17}	0.00×10^{-17}
Hybrid(n)			—	0.00×10^{-17}	0.00×10^{-17}	0.00×10^{-17}
Gersh				—	0.00×10^{-17}	0.00×10^{-17}
LH					—	0.00×10^{-17}
Fixed						—
Mean	0.277	0.303	0.256	0.219	0.472	0.413

TABLE 6.18: *Differences in respect of the mean number of stops obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Normalised mean number of stops						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	9.48×10^{-14}	2.93×10^{-9}	0.00×10^{-17}	0.00×10^{-17}	0.00×10^{-17}
O-TSCA(n)		—	0.00×10^{-17}	0.00×10^{-17}	0.00×10^{-17}	0.00×10^{-17}
Hybrid(n)			—	0.00×10^{-17}	0.00×10^{-17}	0.00×10^{-17}
Gersh				—	0.00×10^{-17}	0.00×10^{-17}
LH					—	0.00×10^{-17}
Fixed						—
Mean	0.087	0.095	0.082	0.069	0.146	0.128

TABLE 6.19: *Differences in respect of the normalised mean number of stops obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Mean time spent travelling under 10km/h						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	0.00×10^{-17}	5.97×10^{-12}	2.30×10^{-1}	0.00×10^{-17}	0.00×10^{-17}
O-TSCA(n)		—	5.51×10^{-4}	0.00×10^{-17}	0.00×10^{-17}	0.00×10^{-17}
Hybrid(n)			—	4.37×10^{-9}	0.00×10^{-17}	0.00×10^{-17}
Gersh				—	0.00×10^{-17}	0.00×10^{-17}
LH					—	0.00×10^{-17}
Fixed						—
Mean	14.34	13.40	13.70	14.24	19.15	16.18

TABLE 6.20: *Differences in respect of the mean time spent travelling under 10km/h obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular* 3×4 *grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Normalised mean time spent travelling under 10km/h						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	0.00×10^{-17}	7.44×10^{-12}	9.07×10^{-3}	0.00×10^{-17}	0.00×10^{-17}
O-TSCA(n)		—	1.21×10^{-2}	1.35×10^{-11}	0.00×10^{-17}	0.00×10^{-17}
Hybrid(n)			—	5.03×10^{-6}	0.00×10^{-17}	0.00×10^{-17}
Gersh				—	0.00×10^{-17}	0.00×10^{-17}
LH					—	0.00×10^{-17}
Fixed						—
Mean	0.1017	0.0964	0.0978	0.1003	0.1308	0.1135

TABLE 6.21: *Differences in respect of the normalised mean time spent travelling under 10km/h obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular* 3×4 *grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

The O-TSCA(n) and Hybrid(n) outperformed the other four algorithms in terms of the mean time a vehicle spends travelling under 10km/h and the normalised mean time a vehicle spends travelling under 10km/h, achieving similar PMI values (see Figures 6.5(e) and 6.5(f) as well as Tables 6.20 and 6.21). LH and Fixed performed significantly worse than the other four algorithms with respect to the two aforementioned PMIs.

According to Einhorn [16], Hybrid performed the best in terms of mean delay time and normalised mean delay time, while in this case it was found that O-TSCA(n) performed the best, followed by Hybrid(n). In terms of mean number of stops and normalised mean number of stops it was found by Einhorn [16] that the O-TSCA achieved the smallest values, while in this study it was found that Gersh performed the best in this respect.

There is not an obvious superior algorithm in this specific scenario as there was not a single algorithm that performs the best over all six PMIs, and the PMIs cannot be ranked according to importance as the relative importance of the PMIs is subjective to road users. It can, however, be claimed that LH was statistically the worst performing algorithm in a regular 3×4 grid of intersections under lighter traffic conditions. The other four algorithms performed comparatively well, but the O-TSCA(n) and Gersh obtained slightly better results than the I-TSCA(n) and Hybrid(n) overall, and may thus be considered the best performing algorithms in the context of a 3×4 grid of intersections under lighter traffic conditions.

Simulation results obtained under heavier traffic conditions ($\lambda = 20$ vehicles/min)

Once again the ANOVA column in Table 6.22 indicates that for all six PMIs there are statistical differences between the PMI means returned by the algorithms at a 5% level of significance. Furthermore, the results of the Levene tests revealed that the variances of the mean samples are statistically distinguishable for all six of the PMIs. Therefore the Games-Howell *post hoc* test was performed to determine between which pairs of algorithmic outputs differences between the PMI means are discernible in respect of all the PMIs.

	Mean value						p-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	ANOVA	Levene's Test
1	66.92	68.49	66.52	77.90	66.41	57.97	$<1 \times 10^{-17}$	1.52×10^{-5}
2	1.716	1.731	1.714	1.834	1.711	1.630	$<1 \times 10^{-17}$	6.63×10^{-6}
3	1.661	1.762	1.762	3.359	1.637	1.745	$<1 \times 10^{-17}$	3.57×10^{-5}
4	0.522	0.547	0.557	1.038	0.516	0.575	$<1 \times 10^{-17}$	1.17×10^{-5}
5	35.38	36.67	34.76	44.64	35.37	28.26	$<1 \times 10^{-17}$	9.45×10^{-6}
6	0.2038	0.2113	0.2015	0.2359	0.2065	0.1682	$<1 \times 10^{-17}$	3.89×10^{-6}

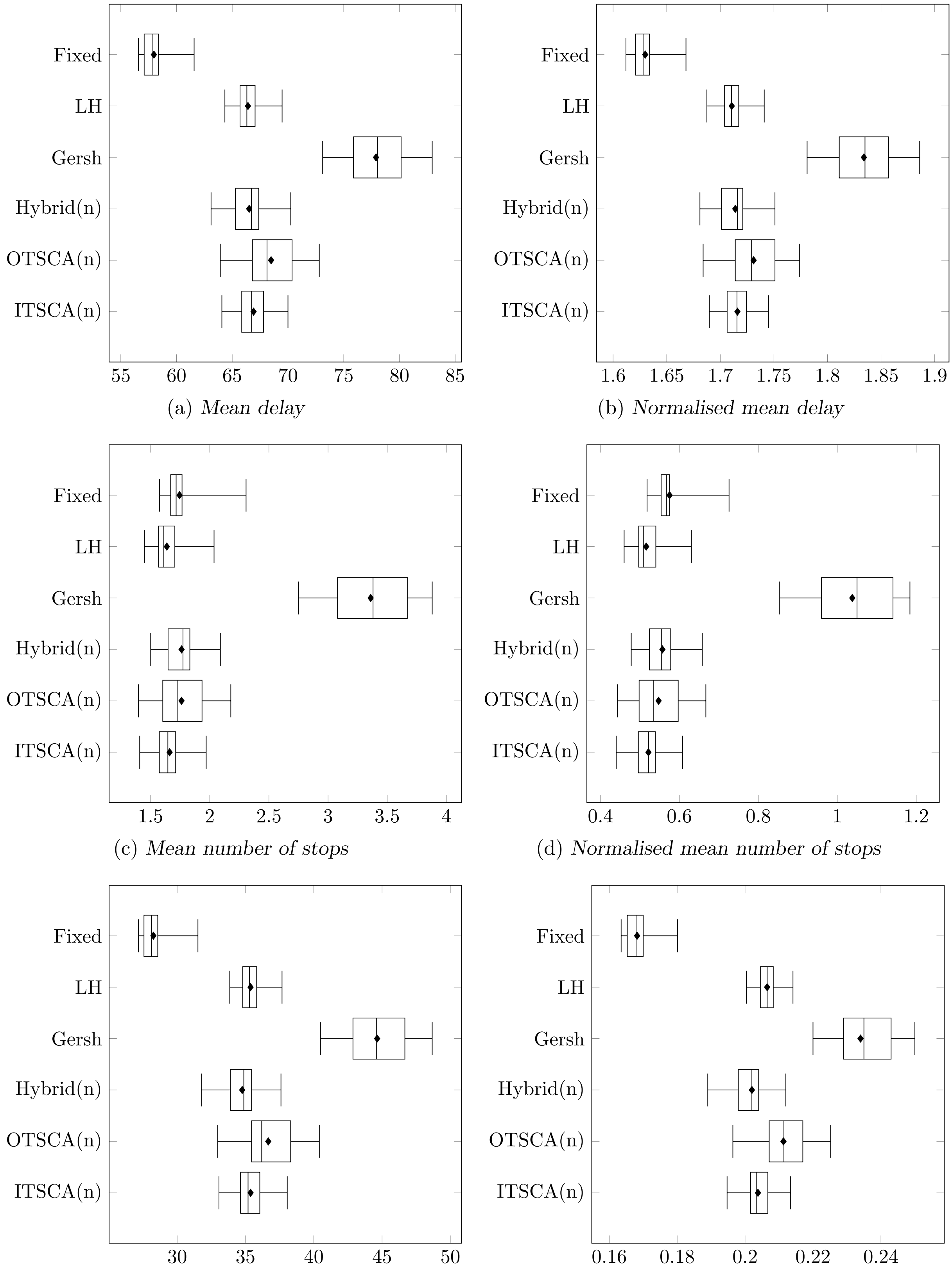
TABLE 6.22: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under heavier traffic conditions in a 3×4 grid of intersections. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

Fixed was found to perform very well in terms of mean delay time and normalised mean delay time, achieving values of 57.96 seconds and 1.63, respectively, while it was not outperformed by any other algorithms in this respect. LH, Hybrid(n) and the I-TSCA(n) were the next best performing algorithms in this respect, outperforming both O-TSCA(n) and Gersh and not differing significantly from one another. Gersh did not perform as well comparatively which is clear from Figures 6.6(a) and 6.6(b), and Tables 6.23 and 6.24.

In respect of the mean and normalised mean number of stops, all the algorithms perform relatively similar, with the exception of Gersh, which was outperformed by all other algorithms.

As was the case for the mean delay time PMIs, Fixed was the best performing algorithm in respect of the mean and normalised mean time vehicles spend travelling under 10km/h, obtaining values of 28.26 seconds and 0.1682, respectively. The next best performing algorithms in this regard were Hybrid(n), I-TSCA(n) and LH whose performances do not differ significantly from one another.

It is concluded that Gersh is the overall worst performing algorithm in the context of a regular 3×4 grid of intersections under heavier traffic conditions, obtaining the largest values across all six PMIs, as seen in Figure 6.6. This is as a result of the long green times that Gersh employed, as it struggled to meet the switching criteria of rule 4 (on which it relies), and which worked effectively under lighter traffic conditions. It was found for all six PMIs that the I-TSCA(n), Hybrid(n) and LH did not differ statistically from another, while the O-TSCA(n) was statistically outperformed in terms of all PMIs other than the mean number of stops and normalised mean number of stops. In conclusion, the I-TSCA(n), Hybrid(n) and LH are all seen as statistically better than the O-TSCA(n), although not statistically different from one another. Fixed achieved the most favourable results overall and is therefore the algorithm best suited in this grid network topology under heavier traffic conditions.



(a) *Mean delay*

(b) *Normalised mean delay*

(c) *Mean number of stops*

(d) *Normalised mean number of stops*

(e) *Mean time spent travelling under 10km/h*

(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 6.6: *PMI results for the five self-organising algorithms and the fixed-time control strategy in the context of a regular* 3×4 *grid of intersections under heavier traffic conditions.*

p-values of the Games-Howell test: Mean delay						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	2.52×10^{-2}	9.21×10^{-1}	1.10×10^{-12}	6.05×10^{-1}	0.00×10^{-17}
O-TSCA(n)		—	5.01×10^{-3}	8.57×10^{-12}	6.98×10^{-4}	4.67×10^{-12}
Hybrid(n)			—	0.00×10^{-17}	9.99×10^{-1}	1.39×10^{-11}
Gersh				—	0.00×10^{-17}	0.00×10^{-17}
LH					—	0.00×10^{-17}
Fixed						—
Mean	66.92	68.49	66.52	77.90	66.41	57.96

TABLE 6.23: *Differences in respect of the mean delay obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular 3 × 4 grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean delay						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	3.41×10^{-2}	9.88×10^{-1}	9.01×10^{-13}	5.45×10^{-1}	0.00×10^{-17}
O-TSCA(n)		—	1.73×10^{-2}	4.64×10^{-12}	9.11×10^{-4}	7.74×10^{-12}
Hybrid(n)			—	0.00×10^{-17}	9.76×10^{-1}	1.39×10^{-11}
Gersh				—	0.00×10^{-17}	0.00×10^{-17}
LH					—	0.00×10^{-17}
Fixed						—
Mean	1.716	1.731	1.714	1.834	1.711	1.630

TABLE 6.24: *Differences in respect of the normalised mean delay obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular 3 × 4 grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	2.40×10^{-1}	7.19×10^{-2}	0.00×10^{-17}	9.76×10^{-1}	2.67×10^{-1}
O-TSCA(n)		—	9.99×10^{-1}	0.00×10^{-17}	7.02×10^{-2}	9.48×10^{-1}
Hybrid(n)			—	5.89×10^{-13}	8.90×10^{-3}	9.38×10^{-1}
Gersh				—	0.00×10^{-17}	2.95×10^{-12}
LH					—	1.50×10^{-1}
Fixed						—
Mean	1.661	1.762	1.762	3.359	1.637	1.745

TABLE 6.25: *Differences in respect of the mean number of stops obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular 3 × 4 grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	3.69×10^{-1}	1.98×10^{-2}	0.00×10^{-17}	9.91×10^{-1}	1.46×10^{-2}
O-TSCA(n)		—	9.77×10^{-1}	0.00×10^{-17}	1.53×10^{-1}	2.38×10^{-1}
Hybrid(n)			—	7.97×10^{-13}	2.93×10^{-3}	4.07×10^{-1}
Gersh				—	0.00×10^{-17}	6.53×10^{-12}
LH					—	7.24×10^{-3}
Fixed						—
Mean	0.521	0.547	0.557	1.038	0.516	0.575

TABLE 6.26: *Differences in respect of the normalised mean number of stops obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular 3 × 4 grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean time spent travelling under 10km/h						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	2.29×10^{-2}	3.87×10^{-1}	8.35×10^{-13}	9.99×10^{-1}	1.14×10^{-12}
O-TSCA(n)		—	4.92×10^{-4}	5.81×10^{-12}	1.53×10^{-2}	1.25×10^{-11}
Hybrid(n)			—	0.00×10^{-17}	3.06×10^{-1}	6.58×10^{-12}
Gersh				—	0.00×10^{-17}	0.00×10^{-17}
LH					—	1.14×10^{-12}
Fixed						—
Mean	35.38	36.67	34.76	44.64	35.37	28.26

TABLE 6.27: *Differences in respect of the mean time spent travelling under 10km/h obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean time spent travelling under 10km/h						
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed
I-TSCA(n)	—	9.30×10^{-5}	4.52×10^{-1}	8.87×10^{-13}	5.68×10^{-2}	5.00×10^{-12}
O-TSCA(n)		—	2.60×10^{-6}	8.81×10^{-12}	1.77×10^{-2}	0.00×10^{-17}
Hybrid(n)			—	0.00×10^{-17}	1.38×10^{-3}	1.71×10^{-11}
Gersh				—	0.00×10^{-17}	0.00×10^{-17}
LH					—	0.00×10^{-17}
Fixed						—
Mean	0.2038	0.2113	0.2015	0.2359	0.2065	0.1682

TABLE 6.28: *Differences in respect of the normalised mean time spent travelling under 10km/h obtained for each of the five self-organising algorithms and the fixed-time control strategy in the context of a regular 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

6.4 Chapter summary

This chapter opened with a summary of the parameter values chosen for the algorithms in §6.1. This was followed in §6.2 by repeating an experiment previously carried out by Einhorn [16], which included a four-intersection corridor. The results revealed that under lighter traffic conditions there was not a single statistically superior algorithm for the four-intersection corridor at a 95% level of confidence, while Fixed and O-TSCA(n) were found to be the best performing algorithms under heavier traffic conditions. In §6.3, another one of Einhorn's experiments was reproduced, this time involving a 3×4 grid of intersections. In this topology, it was found that the O-TSCA(n), Gersh and Hybrid(n) obtained slightly, yet significantly, better results than the other algorithms under lighter traffic conditions, while LH and Fixed were found to be the superior algorithms under heavier traffic conditions.

CHAPTER 7

New self-organising algorithms

Contents

Two novel self-organising traffic signal control algorithms are proposed in this chapter. The algorithm introduced in §7.1 operates by partitioning incoming vehicles into a number of platoons, based on the proximity of vehicles to one another. The second self-organising algorithm is introduced in §7.2 and uses road saturation levels of conflicting traffic flows to make signal switching decisions. These two algorithms are designed in such a way that the major downfalls identified in the previous five self-organising algorithms of Chapter 5 are avoided. This is followed by a comparison of the results returned by the new algorithms with those of their counterparts of Chapter 5 in the context of a four-intersection corridor as well as a 3×4 grid of intersections in §7.3. The chapter finally closes with a brief summary of its contents in §7.4.

7.1 An algorithm based on vehicle platoons

The newly proposed *Vehicle Platoon Traffic Signal Control Algorithm* (VP-TSCA) is a self-organising, adaptive algorithm that clusters vehicles along an intersection approach into a number of groups called *platoons*. The algorithm attempts to switch traffic signals so as not to separate platoons of vehicles travelling through an intersection by switching signals only when the distance between consecutive vehicles exceeds a certain threshold. This algorithm is not predictive, but instead makes use of local, real-time information pertaining to the relevant intersection. The current inter-vehicle threshold distance separating platoons as well as the number of platoons of vehicles on an intersection approach are displayed in the simulation model screen shot of the algorithmic implementation in Figure 7.1, with each colour representing a different platoon.

The *level of traffic congestion* on a road segment is defined as the proportion of the road segment length actually occupied by vehicles. This level of traffic congestion is denoted by $x \in [0, 1]$. Furthermore, a so-called *within-platoon threshold distance*, denoted by D, is defined as the largest

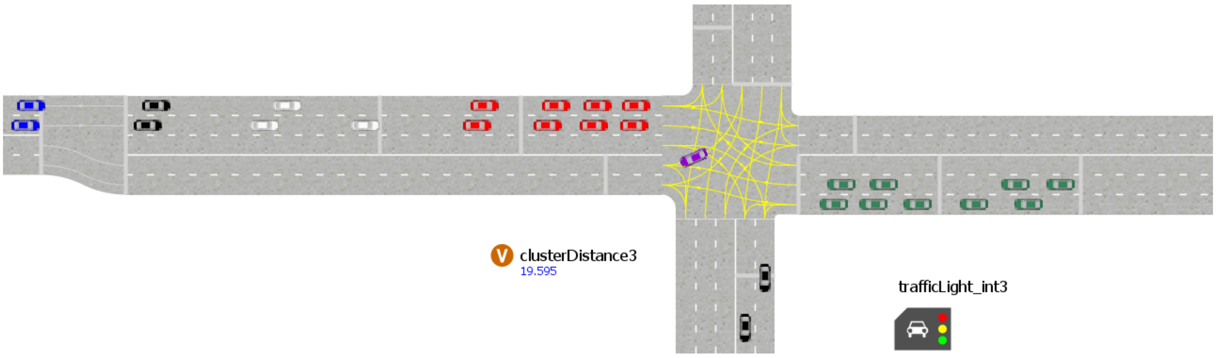


FIGURE 7.1: *A screen shot of the variables employed in the VP-TSCA as implemented in the Anylogic simulation model [3]. Vehicles of the same colour form part of the same platoon, while the within-clustering threshold distance is given by the variable clusterDistance3.*

possible distance between two vehicles which admits platoon formation or clustering in which the two vehicles occupy the same platoon or cluster. The variables x and D are expected to be related to one another in an inversely proportional manner, because:

1. Lighter traffic conditions should result in the adoption of large within-platoon threshold distances as vehicles are more sparsely located along a road segment under such conditions, in which case it should be advantageous to cluster them in order to promote the formation of a platoon.

2. Heavier traffic conditions should result in the adoption of small within-platoon threshold distances in order to avoid the formation of extremely large platoons which may, in turn, give rise to excessively long waiting times for conflicting traffic streams.

Perhaps the simplest functional form for such an inversely proportional relationship is

$$D = \frac{a}{x} + b, \tag{7.1}$$

where $a > 0$ is a constant of proportionality and $b \geq 0$ is an offset constant. Assuming this relationship, it follows that $D \to \infty$ as $x \to 0$. This would seem reasonable, since enforcing a finite upper bound on the platoon formation distance between vehicles seems unnecessary in the case where there are virtually no vehicles present on a road segment, as they should all form part of the same platoon. If, on the other hand, $x \to 1$, then it follows that $D \to a + b$. The purpose of the offset constant b is to facilitate the imposition of a finite lower bound $a + b$ on the within-platoon threshold distance under heavy traffic conditions which is independent of the rate a of the proportional decrease in D as x increases. Extensive numerical simulation experiments carried out by the author have suggested that good values of the constants in (7.1) are $a = 2$ metres and $b = 10$ metres. Thus, the relationship between the within-clustering threshold distance and level of traffic congestion becomes

$$D = \frac{2}{x} + 10. \tag{7.2}$$

The VP-TSCA comprises three main components. The first is a spillback prevention mechanism which ensures that vehicle queues never back up into neighbouring intersections. This is enforced by monitoring slow-moving vehicles along the intersection exit lanes that are within close proximity to the intersection. Let these vehicles belong to the set E_m, where m is the phase that is currently being served.

The second component of the algorithm involves the determination of the level of traffic congestion and thus the within-platoon threshold distance. The level of traffic congestion for a signal phase m is denoted by x_m and is calculated by dividing the sum of the effective road length occupied by vehicles along approaches served during phase m, by the total length of the road L. Thus,

$$x_m = \sum_{j \in C_m} \hat{\ell}_j / L, \tag{7.3}$$

where C_m denotes the ordered set of approaching vehicles and $\hat{\ell}_j$ denotes the length of an approaching vehicle j. Once the road saturation x_m has been determined, the platoon threshold distance may then be calculated according to (7.2).

The third component of the algorithm is concerned with partitioning vehicles into platoons during a phase m. At the start of a phase m, a set P_m is defined to contain all the vehicles that form the initial platoon to be served. A pseudo-code description of the VP-TSCA is presented as Algorithm 7.1.

Algorithm 7.1: The Vehicle Platoon Traffic Signal Control Algorithm

Input : The physical lengths and speeds of all vehicles approaching the intersection, as well as their distances to the intersection, at each time step.

Output: A signal phase switching decision for each time step.

1 **for** *each time step* t **do**
2   **for** *current signal phase* m **do**
3   **if** $|E_m| > 0$ **then**
4   switch signals
5   **if** *elapsed time of signal* $< \epsilon$ **then**
6             $x_m \leftarrow \sum_{j \in C_m} \hat{\ell}_j / L$;
7             $D_m \leftarrow \frac{2}{x_m} + 10$;
8   **for** *each vehicle* j *in* C_m **do**
9   **if** $\sqrt{(j_x - (j-1)_x)^2 + (j_y - (j-1)_y)^2} \leq D_m$ **then**
10                     $P_m \cup \{j\}$
11   **if** $|P_m| = 1$ **then**
12   **for** *each vehicle* j *in* C_m **do**
13   **if** $\sqrt{(j_x - (j-1)_x)^2 + (j_y - (j-1)_y)^2} \leq D_m$ **then**
14                     $P_m \cup \{j\}$
15   **if** $|P_m| = 0$ **then**
16   switch signals;

If, at any point in time, there is a back-up of vehicles from a neighbouring downstream intersection, the relevant signal will be terminated as in lines 3 and 4. At the very start of a signal phase m, the current road saturation and within-clustering threshold distance are calculated in lines 6 and 7, respectively. The within-clustering threshold distance is determined only once per phase duration and this calculation occurs within a short time ϵ from the start of a signal phase. This within-clustering threshold distance is then used to determine the initial platoon to be served in lines 9 and 10. In line 9, the x and y coordinates of a vehicle j are denoted by j_x and j_y, respectively. The size of the initial platoon is continuously monitored. In line 11, a test is performed to determine whether the size of the first platoon to be served is 1. If this is the case, the clustering distance is once again used to determine whether any other vehicles form a platoon together with the last detected vehicle in line 13. If there are indeed vehicles that are considered to be platooned with the last vehicle, they are added to the set P_m in line 14. If there are no further vehicles in P_m, a signal switch is triggered, as shown in lines 15 and 16.

7.2 An algorithm based on road saturation ratios

The *Saturation Ratio Traffic Signal Control Algorithm* (SR-TSCA) is proposed as a new self-organising algorithm for switching traffic signals based on the saturation ratio of current and competing traffic flows. The SR-TSCA, like the VP-TSCA, makes use of real-time kinematic vehicle data available at the pertinent intersection.

The algorithm also employs the notion of road saturation x as a measure of the level of traffic congestion of vehicles served during a certain phase, as well as a *current saturation ratio*, denoted by S. The current saturation ratio is calculated by dividing the road saturation of the phase that is currently receiving service by the road saturation of the competing phase. Furthermore, a *saturation ratio threshold* parameter, denoted by T, is introduced, which is used to trigger a signal change once the current saturation ratio falls below this threshold. The road saturation is once again expected to be inversely proportional to the threshold parameter in this instance, since:

1. Effective traffic signal switching strategies require relatively frequent switching of signals under lighter traffic conditions. This results in shorter green times which are associated with a larger saturation ratio threshold.

2. Heavier traffic conditions should result in the adoption of longer green times which are achieved by employing smaller saturation ratio thresholds.

The general relationship between x and T may thus be expressed by

$$T = \frac{c}{x} + d, \tag{7.4}$$

where $c > 0$ is a constant of proportionality and $d \geq 0$ is an offset constant. It follows from (7.4) that $T \to \infty$ as $x \to 0$. Under very light traffic conditions this could result in a current saturation ratio that immediately exceeds the threshold value, thus triggering a signal change. This does not, however, pose a problem as an immediate signal change is prevented by enforcing a minimum green time. If, on the other hand, $x \to 1$, then it follows from (7.4) that $T \to c + d$. An appropriate minimum saturation ratio threshold is necessary, and was ultimately chosen to be 0.3 through extensive simulation experimentation. This indicates that a phase will be served until the ratio of the current phase demand to the competing phase demand is at the very lowest 3 : 10, after which signals will change, provided that the minimum green time has elapsed. Through extensive experimentation, suitable values for the constants in (7.4) were found to be $c = \frac{1}{15}$ and $d = \frac{7}{30}$. The relationship between x and T therefore becomes

$$T = \frac{1}{15x} + \frac{7}{30}. \tag{7.5}$$

Like the VP-TSCA, this algorithm also comprises three main components. The first component is the same spillback prevention mechanism described in §7.1. Through observation of numerous simulation runs of the other self-organising algorithms of Chapter 5, it was found that vehicle back-ups into neighbouring intersections are detrimental to the performance of the algorithms. As a result, the spill-back mechanism was put in place in both of the newly proposed algorithms of this chapter in order to avoid this phenomenon.

The second component of the SR-TSCA involves the calculation of a suitable minimum green time, denoted by $g_{\min,m}$ for a phase m. This green time is not fixed, but instead varies depending on the number of vehicles currently queued at the intersection at the start of a green phase. Let Q_i denote the set of vehicles queued in lane $i \in I$, where I is the set of all approach lanes served during phase m. The minimum green time is calculated (offline) as the time required to clear the size of the longest queue over all lanes in I served during phase m present at the start of the green phase, denoted by $Q_{max,m}$, such that every initially detected, queued vehicle at the intersection can be served during that phase. The minimum green time is calculated as

$$g_{\min,m} = 2.5 + 2(Q_{max,m} - 1). \tag{7.6}$$

If queues are exceptionally large, it is not problematic, as the vehicle detection equipment is assumed only to be able to detect up to 275 metres upstream of an intersection. As a result, only the 275 metre approach is considered when determining the suitable minimum green time.

The third and final component of the algorithm is its signal switching strategy, which is based on the current saturation ratio and the calculated saturation ratio threshold.

A screen shot of the variable values of this algorithm, as implemented in Anylogic [3], is depicted in Figure 7.2. The variable *minGT_HnewAlg2* is the time required to clear the largest queue (five vehicles in this case) in the horizontal direction, while *saturationRatio2* tracks the current saturation ratio of the phase currently receiving service (the horizontal direction in this scenario) to that of the competing phase (the vertical direction in this scenario). The saturation ratio threshold (*thresholdRatio2*) is calculated according to the function in (7.5) at the start of the signal phase.

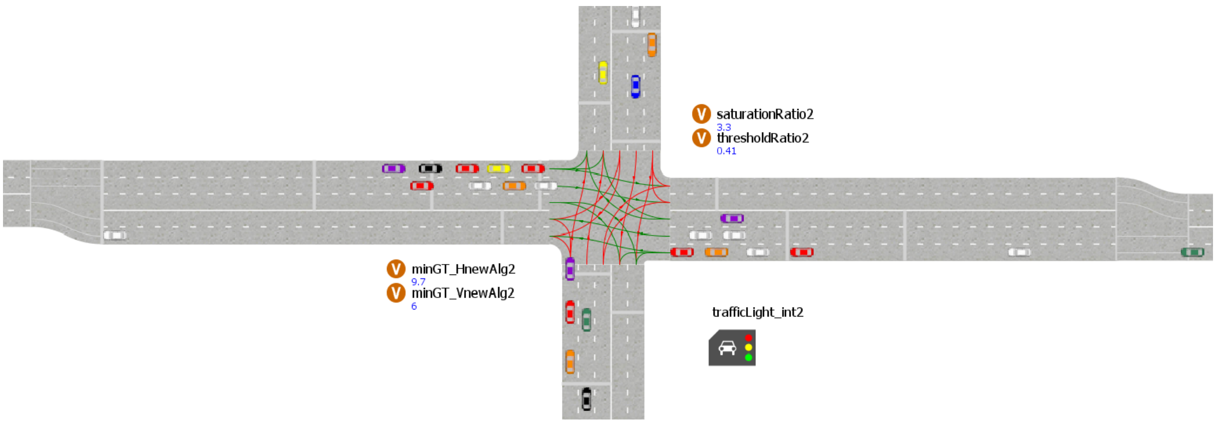


FIGURE 7.2: *A screen shot of the SR-TSCA implemented in the simulation model in Anylogic [3].*

A pseudo-code description of the SR-TSCA is presented as Algorithm 7.2. In lines 3 and 4, the saturation of the current and competing phases of traffic flow are calculated, respectively. The ratio of these two values is then calculated in line 5. A test is performed for vehicle spillback from neighbouring intersections in line 6. The saturation ratio threshold is determined at the start of the current phase in line 9, while the length of the longest queue is determined in lines 10–13. The minimum green time for the phase m is calculated in line 14, and this value is used

Algorithm 7.2: The Saturation Ratio Traffic Signal Control Algorithm

Input : The physical lengths and speeds of all vehicles approaching the intersection, as well as their distances to the intersection, at each time step.

Output: A signal phase switching decision for each time step.

for *each time step* t **do**
    for *current signal phase* m *and competing phase* m' **do**
        $x_m \leftarrow \sum_{j \in C_m} \hat{\ell}_j / L$;
        $x_{m'} \leftarrow \sum_{j \in C_{m'}} \hat{\ell}_j / L$;
        $S_m \leftarrow \frac{x_m}{x_{m'}}$;
        if $|E_m| > 0$ **then**
            switch signals
        if *elapsed time of signal* $< \epsilon$ **then**
            $T_m \leftarrow \frac{1}{15 x_m} + \frac{7}{30}$;
            $Q_{max} = 0$;
            for *int* i *:* I **do**
                if $|Q_i| > Q_{max}$ **then**
                    $Q_{max} = |Q_i|$;
            $g_{min,m} = 2.5 + 2(Q_{max} - 1)$;
        if *elapsed time of signal* $> g_{min,m}$ **then**
            if $S_m \leq T_m$ **then**
                switch signals

in line 15 to determine whether the minimum green time has elapsed. The current saturation ratio is finally compared with the saturation ratio threshold in line 16. If the current saturation ratio is exceeded, a signal switch is triggered in line 17.

7.3 Algorithmic performance

In this section, the performances of the newly proposed VP-TSCA and SR-TSCA are compared with one another as well as with those of LH, Gersh, Fixed and the three improved algorithms of §5.3. The comparisons take place in the context of a road corridor (§7.3.1) as well as in a gridded road network (§7.3.2). Once again a warm-up period of 1 800 seconds is imposed for each simulation run, for a total of 30 simulation replications. A Levene test [67] is used in order to determine whether the variances of the algorithmic means are significantly different. If this is the case, the Games-Howell test [19] is used to determine where these differences lie. If, on the other hand, there is no significant difference between variances, the Fisher LSD test [85] is used to determine where the differences lie.

7.3.1 Algorithmic performance comparison in a corridor network

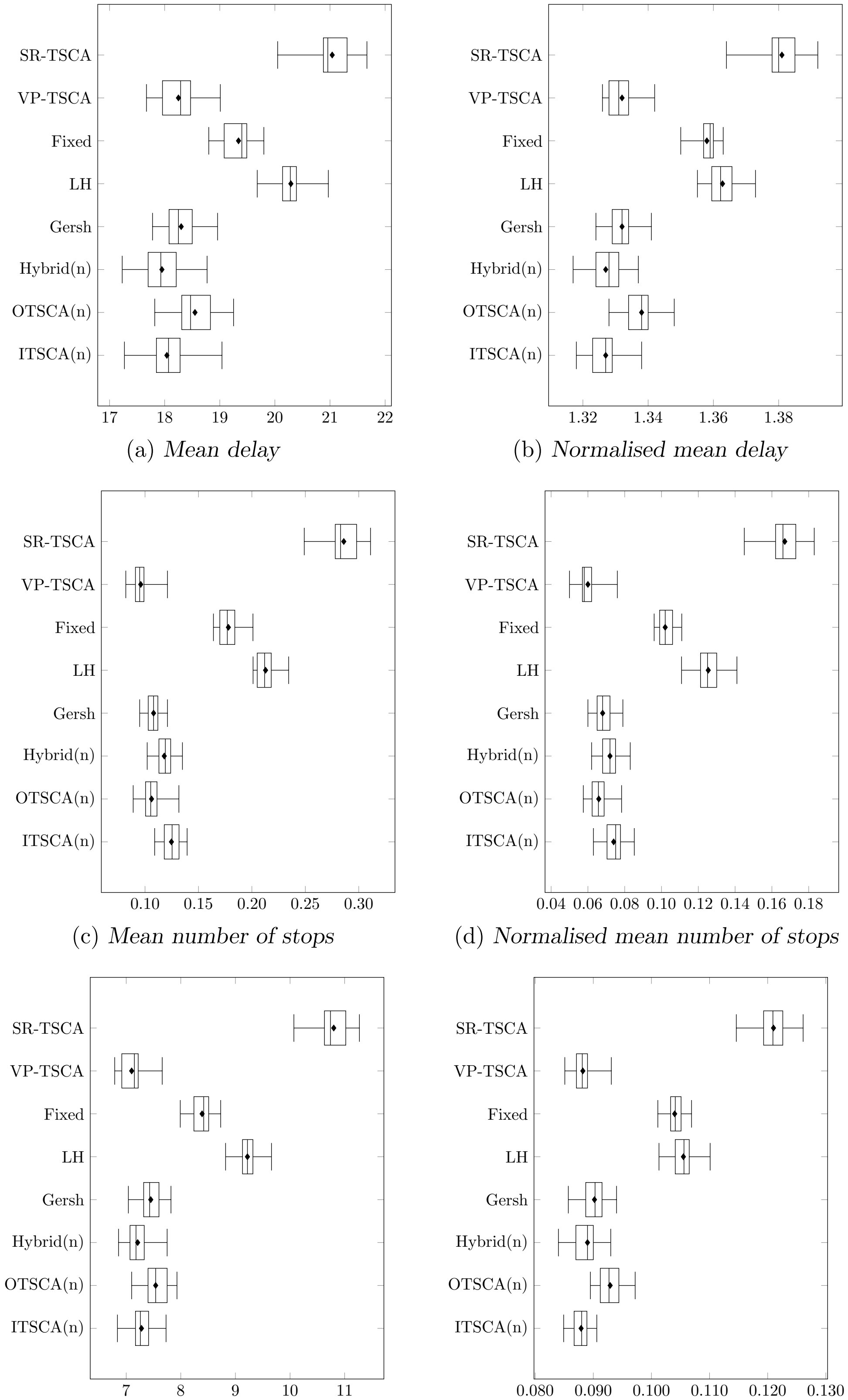
The relative performances of the eight algorithms (the two new algorithms of §7 and the six existing algorithms of §5) are compared in this section with respect to the six PMIs of §4.1.4 in a four-intersection corridor road network under both lighter and heavier traffic conditions.

Simulation results for lighter traffic conditions ($\lambda = 10$ vehicles per minute)

The ANOVA column in Table 7.1 indicates that for all six PMIs there are statistical differences between the means returned by the algorithms at a 5% level of significance. Furthermore, the outcome of the Levene test revealed that the algorithmic variances of the mean samples were statistically indistinguishable for PMI 1, PMI 2 and PMI 5, and therefore the Fisher LSD *post hoc* test was used in order to determine between which pairs of algorithmic outputs differences are statistically discernible in respect of these three PMIs. The Games-Howell *post hoc* test was used for this purpose in respect of the remaining three PMIs, namely PMI 3, PMI 4 and PMI 6 as the test indicated that there was a significant difference between the variances of the sample means for these PMIs at a 5% level of significance.

	Mean value								p-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA	ANOVA	Levene's Test
1	18.04	18.55	17.95	18.30	20.29	19.34	18.25	21.10	$<1\times10^{-17}$	5.81×10^{-1}
2	1.327	1.338	1.327	1.332	1.363	1.358	1.331	1.382	$<1\times10^{-17}$	1.18×10^{-1}
3	0.125	0.106	0.118	0.108	0.213	0.178	0.096	0.286	$<1\times10^{-17}$	8.82×10^{-4}
4	0.074	0.066	0.072	0.068	0.125	0.102	0.059	0.167	$<1\times10^{-17}$	1.29×10^{-3}
5	7.28	7.54	7.21	7.45	9.22	8.39	7.10	10.83	$<1\times10^{-17}$	7.82×10^{-1}
6	0.0879	0.0929	0.0885	0.0902	0.1055	0.1040	0.0882	0.1216	$<1\times10^{-17}$	4.27×10^{-2}

TABLE 7.1: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under lighter traffic conditions in a four-intersection corridor. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*



(a) *Mean delay*

(b) *Normalised mean delay*

(c) *Mean number of stops*

(d) *Normalised mean number of stops*

(e) *Mean time spent travelling under 10km/h*

(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 7.3: *PMI results for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 within the context of a four-intersection corridor under lighter traffic conditions.*

p-values of the Fisher LSD test: Mean delay								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.90×10^{-8}	2.66×10^{-1}	3.08×10^{-3}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	2.02×10^{-2}	$<1 \times 10^{-17}$
O-TSCA(n)		—	3.89×10^{-11}	5.00×10^{-3}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	5.88×10^{-4}	$<1 \times 10^{-17}$
Hybrid(n)			—	5.59×10^{-5}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	6.56×10^{-4}	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	5.16×10^{-1}	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Fixed						—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	18.04	18.55	17.95	18.30	20.29	19.34	18.25	21.10

TABLE 7.2: *Differences in respect of the mean delay obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a four-intersection corridor. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Normalised mean delay								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	$<1 \times 10^{-17}$	5.55×10^{-1}	2.54×10^{-5}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	4.18×10^{-5}	$<1 \times 10^{-17}$
O-TSCA(n)		—	3.33×10^{-16}	8.32×10^{-7}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	4.64×10^{-7}	$<1 \times 10^{-17}$
Hybrid(n)			—	2.63×10^{-4}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	4.10×10^{-4}	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	9.04×10^{-1}	$<1 \times 10^{-17}$
LH					—	1.17×10^{-4}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Fixed						—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	1.327	1.338	1.327	1.332	1.363	1.358	1.331	1.382

TABLE 7.3: *Differences in respect of the normalised mean delay obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a four-intersection corridor. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	3.84×10^{-10}	5.22×10^{-2}	8.49×10^{-11}	1.33×10^{-11}	1.41×10^{-11}	1.13×10^{-11}	$<1 \times 10^{-17}$
O-TSCA(n)		—	2.21×10^{-5}	9.92×10^{-1}	1.49×10^{-11}	1.49×10^{-11}	3.26×10^{-3}	$<1 \times 10^{-17}$
Hybrid(n)			—	1.92×10^{-5}	1.20×10^{-11}	1.31×10^{-11}	1.36×10^{-11}	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	4.14×10^{-5}	8.07×10^{-13}
LH					—	1.48×10^{-11}	1.45×10^{-11}	$<1 \times 10^{-17}$
Fixed						—	1.40×10^{-11}	$<1 \times 10^{-17}$
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	0.125	0.106	0.118	0.1078	0.213	0.178	0.096	0.286

TABLE 7.4: *Differences in respect of the mean number of vehicle stops obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	3.73×10^{-6}	6.96×10^{-1}	2.31×10^{-3}	4.52×10^{-12}	1.02×10^{-11}	1.51×10^{-11}	$<1 \times 10^{-17}$
O-TSCA(n)		—	6.91×10^{-4}	5.03×10^{-1}	1.44×10^{-15}	1.32×10^{-11}	7.08×10^{-4}	$<1 \times 10^{-17}$
Hybrid(n)			—	1.66×10^{-1}	$<1 \times 10^{-17}$	1.43×10^{-11}	7.12×10^{-11}	6.04×10^{-13}
Gersh				—	$<1 \times 10^{-17}$	1.44×10^{-11}	5.95×10^{-7}	6.76×10^{-13}
LH					—	$<1 \times 10^{-17}$	8.35×10^{-12}	$<1 \times 10^{-17}$
Fixed						—	6.68×10^{-12}	1.10×10^{-12}
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	0.074	0.066	0.072	0.068	0.125	0.102	0.059	0.167

TABLE 7.5: *Differences in respect of the normalised mean number of stops obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	8.08×10^{-7}	2.24×10^{-1}	9.36×10^{-4}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	1.41×10^{-3}	$<1 \times 10^{-17}$
O-TSCA(n)		—	1.56×10^{-9}	8.70×10^{-2}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	8.44×10^{-15}	$<1 \times 10^{-17}$
Hybrid(n)			—	7.87×10^{-6}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	4.54×10^{-2}	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	3.04×10^{-10}	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Fixed						—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	7.28	7.54	7.21	7.45	9.22	8.39	7.10	10.83

TABLE 7.6: *Differences in respect of the mean time vehicles spent travelling under 10km/h obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	6.04×10^{-12}	8.83×10^{-1}	9.98×10^{-5}	8.79×10^{-13}	1.47×10^{-11}	9.97×10^{-1}	$<1 \times 10^{-17}$
O-TSCA(n)		—	1.76×10^{-10}	2.77×10^{-5}	1.41×10^{-11}	3.70×10^{-12}	1.43×10^{-11}	$<1 \times 10^{-17}$
Hybrid(n)			—	3.35×10^{-2}	1.49×10^{-11}	$<1 \times 10^{-17}$	9.97×10^{-1}	1.32×10^{-12}
Gersh				—	1.44×10^{-11}	2.31×10^{-12}	1.89×10^{-3}	$<1 \times 10^{-17}$
LH					—	2.46×10^{-2}	1.02×10^{-1}	5.72×10^{-13}
Fixed						—	1.01×10^{-11}	$<1 \times 10^{-17}$
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	0.0879	0.0929	0.0885	0.0902	0.1055	0.1040	0.0882	0.1216

TABLE 7.7: *Differences in respect of the normalised mean time vehicles spent travelling under 10km/h obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

The VP-TSCA yielded fairly good results that are similar to those returned by Gersh in respect of mean delay time as well as normalised mean delay time, obtaining values of 18.25 and 1.331 (see Tables 7.2 and 7.3), respectively. The VP-TSCA was outperformed by the I-TSCA(n) and Hybrid(n) in respect of mean delay time, but outperformed all other algorithms aside from Gersh, from which it did not differ significantly. In contrast to the favourable behaviour achieved by the VP-TSCA, the SR-TSCA performed poorly in terms of mean delay time and normalised mean delay time, achieving values of 21.04 and 1.381 (see Tables 7.2 and 7.3), respectively. As can be seen in Figures 7.3(a) and 7.3(b), the SR-TSCA is statistically the worst performing algorithm in respect of these two PMIs at a 95% level of confidence.

The VP-TSCA achieved the smallest value for both mean number of stops and normalised mean number of stops performed by vehicles, statistically outperforming all other algorithms. This indicates that a considerable amount of coordination was achieved between traffic intersections. The VP-TSCA obtained a mean number of stops value of 0.096 and a normalised mean number of stops value of 0.059, an improvement of 9% and 10%, respectively, over the O-TSCA(n), which is the next best performing algorithm (see Tables 7.4 and 7.5). The SR-TSCA, on the other hand, is statistically the worst performing algorithm in respect of these two PMIs as it achieved the largest value for both. This is clear from the box plots in Figures 7.3(c) and 7.3(d).

In respect of mean time spent travelling under 10km/h, the VP-TSCA achieved relatively good results, obtaining a value of 7.10 seconds (see Table 7.6) and outperforming all algorithms but Hybrid(n), from which is does not differ significantly. Once again, the SR-TSCA achieved relatively poor results and is statistically the worst performing algorithm in respect of mean time spent travelling under 10km/h, obtaining a value of 10.80 seconds (see Table 7.6). The VP-TSCA outperformed all other algorithms in terms of normalised mean time spent travelling under 10km/h (see Table 7.7), except for the I-TSCA(n) and Hybrid(n), from which is does not significantly differ. The VP-TSCA achieved a value of 0.0882, indicating that vehicles spend, on average, only 8.82% of their time travelling at speeds slower than 10km/h. The SR-TSCA, on the other hand, achieved a corresponding value of 0.1210 (see Table 7.7), indicating that vehicles spend 12.10% of their time travelling under 10km/h. Once again, the SR-TSCA performed statistically worse than all other algorithms at a 95% level of confidence, which is apparent in the box plots of Figures 7.3(e) and 7.3(f).

Overall, the SR-TSCA was the worst performing algorithm in respect of all six PMIs of §4.1.4 under lighter traffic conditions in a four-intersection corridor. This is due to the longer green times allocated by the SR-TSCA, which are not suitable for lighter traffic densities. The VP-TSCA, on the other hand, achieved promising results over all six PMI values and performed particularly well in respect of mean number of stops and normalised mean number of stops. The success of the VP-TSCA in respect of vehicle stops is attributed to the clustering of vehicles into various platoons, such that signals are switched in a way that the separation of platoons is avoided as far as possible, thus reducing vehicle stops.

Simulation results for heavier traffic conditions ($\lambda = 20$ vehicles per minute)

The ANOVA column in Table 7.8 indicates that for all six PMIs there are statistical differences between the means returned by the algorithms at a 5% level of significance. Furthermore, the outcome of the Levene test revealed that the variances of the mean samples were statistically different for all six of the PMIs at a 5% level of significance. The Games-Howell *post hoc* test was therefore employed in respect of all of the PMIs to determine between which pairs of algorithmic outputs differences are statistically discernible.

	Mean value								p-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA	ANOVA	Levene's Test
1	33.58	33.06	33.00	36.22	33.37	31.27	32.05	32.67	$<1 \times 10^{-17}$	2.86×10^{-2}
2	1.601	1.598	1.594	1.688	1.609	1.599	1.588	1.595	$<1 \times 10^{-17}$	1.47×10^{-3}
3	0.864	0.768	0.864	1.389	0.850	0.957	0.837	1.007	$<1 \times 10^{-17}$	1.85×10^{-5}
4	0.504	0.458	0.510	0.938	0.518	0.632	0.514	0.598	$<1 \times 10^{-17}$	1.47×10^{-6}
5	18.09	17.56	17.47	20.42	18.01	16.46	16.65	18.06	$<1 \times 10^{-17}$	2.49×10^{-3}
6	0.1739	0.1765	0.1707	0.1986	0.1796	0.1735	0.1684	0.1731	$<1 \times 10^{-17}$	7.89×10^{-3}

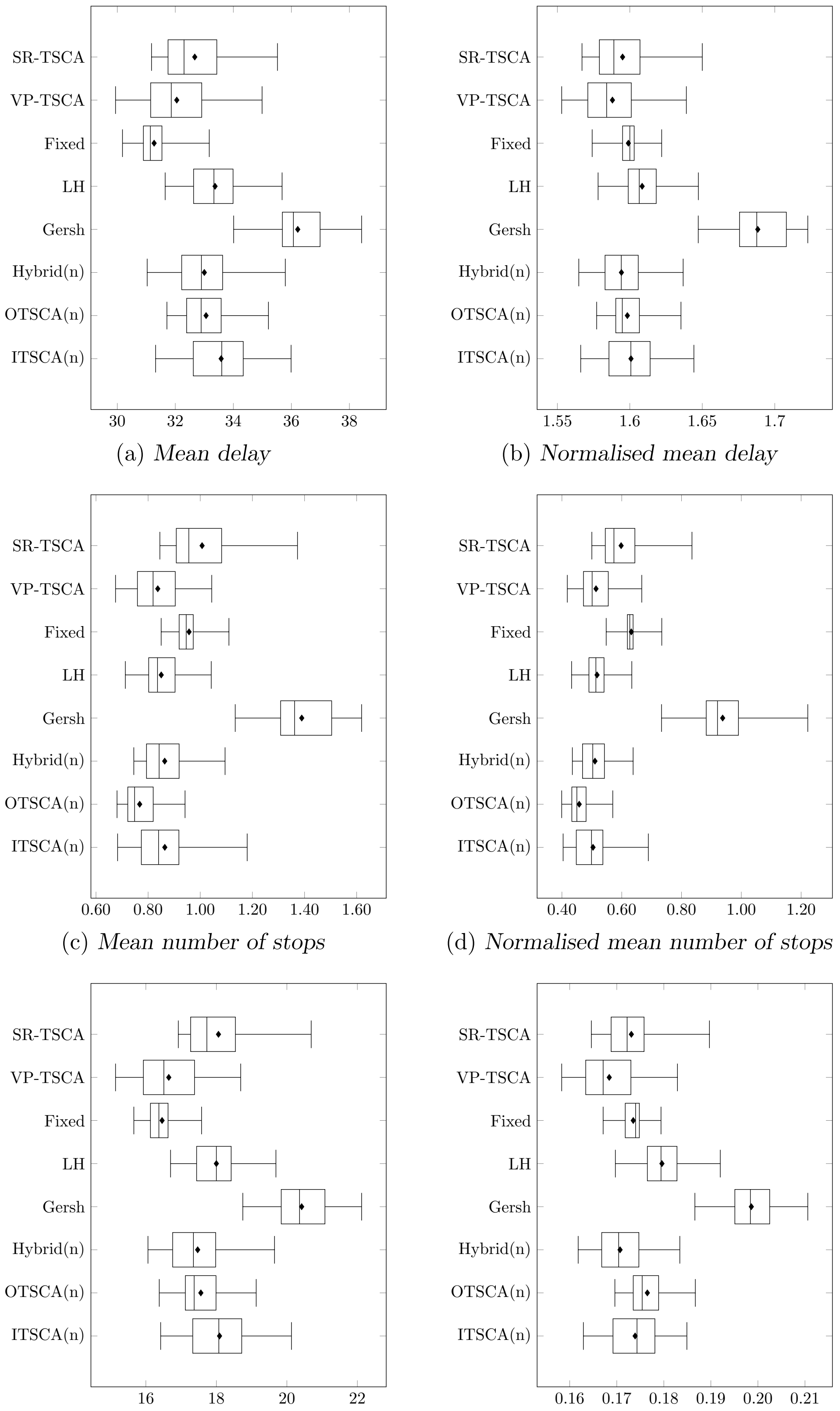
TABLE 7.8: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under heavier traffic conditions in a four-intersection corridor. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

The VP-TSCA achieved a mean delay value of 32.05 seconds, statically outperforming all other algorithms with the exception of Fixed, at a 95% level of confidence (see Table 7.9). While outperformed by the VP-TSCA and Fixed, the SR-TSCA was not outperformed by any another algorithm in respect of mean delay time, although it did not differ significantly from the O-TSCA(n) or Hybrid(n), obtaining a mean delay of 32.67. The VP-TSCA outperformed all algorithms except for Hybrid(n) and the SR-TSCA in terms of normalised mean delay time, not differing significantly from these two algorithms (see Table 7.10). The SR-TSCA achieved a value of 1.595, implying that vehicles spend an additional 59.5% of their time in the road network as a result of delays.

The SR-TSCA struggled to achieve coordination among the intersections as is clear from the mean stops value of 1.007 and normalised mean stops value of 0.598 returned by the algorithm (see Tables 7.11 and 7.12). Furthermore, the SR-TSCA was significantly outperformed by all other algorithms in respect of both these PMI values at a 95% level of confidence, with the exception of Gersh, which was outperformed by the SR-TSCA. The VP-TSCA achieved better results than the SR-TSCA in terms of these PMIs, obtaining a mean number of stops value of 0.837 and a normalised mean stops value of 0.514. The VP-TSCA was outperformed by the O-TSCA(n), and outperformed Fixed, the SR-TSCA and Gersh by 12%, 14% and 45%, respectively.

The VP-TSCA and Fixed achieved the most favourable outcome in respect of mean time spent travelling under 10km/h, obtaining a result of 16.65 and 16.46 seconds, respectively, and significantly outperforming all other algorithms at a 95% level of confidence (see Table 7.13). The SR-TSCA did not perform as well, as it was outperformed by the O-TSCA(n), Hybrid(n), the VP-TSCA and Fixed, while not differing significantly from the I-TSCA(n) and Gersh. The SR-TSCA performed better in terms of normalised mean time spent travelling under 10km/h, significantly outperforming the O-TSCA(n), LH and Gersh, by 2%, 4% and 13%, respectively (see Table 7.14), while not differing significantly from Fixed. The VP-TSCA once again achieved a relatively good result, obtaining a value of 0.1684 in respect of normalised mean time spent under 10km/h, outperforming all algorithms with the exception of Hybrid(n), from which it did not differ significantly.

The VP-TSCA and Fixed appear to be the most promising algorithms overall, as they returned the most favourable results for the six PMIs. Fixed appears to be less effective at achieving coordination (and hence reducing vehicle stops) under heavier traffic conditions, in comparison with the O-TSCA(n) and VP-TSCA. The performance of the SR-TSCA certainly improved with an increase in traffic flow density, which may be seen in Figures 7.4(a), 7.4(b), 7.4(e) and 7.4(f). Coordination, however, is still poor, as is apparent in Figures 7.4(c) and 7.4(f).



(a) *Mean delay*

(b) *Normalised mean delay*

(c) *Mean number of stops*

(d) *Normalised mean number of stops*

(e) *Mean time spent travelling under 10km/h*

(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 7.4: *PMI results for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 within the context of a four-intersection corridor under heavier traffic conditions.*

	p-values of the Games-Howell test: Mean delay							
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	5.70×10^{-1}	5.08×10^{-1}	5.71×10^{-11}	9.95×10^{-1}	3.23×10^{-10}	1.24×10^{-4}	8.94×10^{-2}
O-TSCA(n)		—	9.99×10^{-1}	6.90×10^{-12}	8.84×10^{-1}	5.31×10^{-11}	5.07×10^{-3}	8.17×10^{-1}
Hybrid(n)			—	1.49×10^{-11}	8.24×10^{-1}	1.26×10^{-8}	2.28×10^{-2}	9.42×10^{-1}
Gersh				—	1.15×10^{-11}	$<1 \times 10^{-17}$	1.39×10^{-11}	1.12×10^{-11}
LH					—	$<1 \times 10^{-17}$	1.51×10^{-4}	1.98×10^{-1}
Fixed						—	3.85×10^{-2}	2.86×10^{-5}
VP-TSCA							—	4.34×10^{-1}
SR-TSCA								—
Mean	33.58	33.06	33.00	36.22	33.37	31.27	32.05	32.67

TABLE 7.9: *Differences in respect of the mean delay obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

	p-values of the Games-Howell test: Normalised mean delay							
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	9.99×10^{-1}	8.58×10^{-1}	1.36×10^{-11}	7.10×10^{-1}	9.99×10^{-1}	2.10×10^{-1}	9.50×10^{-1}
O-TSCA(n)		—	9.65×10^{-1}	$<1 \times 10^{-17}$	1.35×10^{-1}	9.99×10^{-1}	2.68×10^{-1}	9.95×10^{-1}
Hybrid(n)			—	2.20×10^{-12}	2.36×10^{-2}	8.22×10^{-1}	8.87×10^{-1}	9.99×10^{-1}
Gersh				—	$<1 \times 10^{-17}$	8.33×10^{-13}	1.25×10^{-11}	1.50×10^{-11}
LH					—	1.53×10^{-1}	8.54×10^{-4}	1.09×10^{-1}
Fixed						—	1.08×10^{-1}	9.61×10^{-1}
VP-TSCA							—	8.98×10^{-1}
SR-TSCA								—
Mean	1.601	1.598	1.594	1.688	1.609	1.599	1.588	1.595

TABLE 7.10: *Differences in respect of the normalised mean delay obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	9.06×10^{-3}	9.99×10^{-1}	1.35×10^{-11}	9.99×10^{-1}	1.24×10^{-2}	9.79×10^{-1}	1.09×10^{-3}
O-TSCA(n)		—	5.46×10^{-4}	1.13×10^{-12}	5.95×10^{-4}	1.37×10^{-11}	3.61×10^{-2}	3.48×10^{-10}
Hybrid(n)			—	$<1 \times 10^{-17}$	9.98×10^{-1}	6.06×10^{-4}	9.57×10^{-1}	1.79×10^{-1}
Gersh				—	1.77×10^{-13}	8.22×10^{-13}	$<1 \times 10^{-17}$	1.47×10^{-11}
LH					—	2.43×10^{-6}	9.98×10^{-1}	1.39×10^{-5}
Fixed						—	1.38×10^{-5}	5.43×10^{-1}
VP-TSCA							—	9.18×10^{-6}
SR-TSCA								—
Mean	0.864	0.768	0.864	1.389	0.850	0.957	0.837	1.007

TABLE 7.11: *Differences in respect of the mean number of vehicle stops obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	3.49×10^{-2}	9.99×10^{-1}	$<1 \times 10^{-17}$	9.84×10^{-1}	1.69×10^{-10}	9.98×10^{-1}	9.02×10^{-5}
O-TSCA(n)		—	1.28×10^{-3}	$<1 \times 10^{-17}$	3.18×10^{-5}	1.49×10^{-11}	1.51×10^{-3}	3.49×10^{-10}
Hybrid(n)			—	1.04×10^{-12}	9.99×10^{-1}	$<1 \times 10^{-17}$	9.99×10^{-1}	8.35×10^{-5}
Gersh				—	1.00×10^{-13}	$<1 \times 10^{-17}$	6.05×10^{-13}	$<1 \times 10^{-17}$
LH					—	6.82×10^{-12}	9.99×10^{-1}	2.00×10^{-4}
Fixed						—	8.08×10^{-11}	3.76×10^{-1}
VP-TSCA							—	2.99×10^{-4}
SR-TSCA								—
Mean	0.504	0.458	0.511	0.938	0.518	0.632	0.514	0.598

TABLE 7.12: *Differences in respect of the normalised mean number of stops obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	2.57×10^{-1}	1.77×10^{-1}	1.29×10^{-11}	9.99×10^{-1}	1.35×10^{-8}	3.35×10^{-6}	9.99×10^{-1}
O-TSCA(n)		—	9.99×10^{-1}	3.17×10^{-12}	2.00×10^{-1}	3.53×10^{-8}	5.36×10^{-4}	3.33×10^{-1}
Hybrid(n)			—	1.46×10^{-11}	1.43×10^{-1}	1.25×10^{-5}	6.84×10^{-3}	2.33×10^{-1}
Gersh				—	5.79×10^{-12}	3.14×10^{-13}	1.49×10^{-11}	1.17×10^{-11}
LH					—	0.00×10^{-17}	1.83×10^{-7}	9.99×10^{-1}
Fixed						—	9.62×10^{-1}	2.26×10^{-8}
VP-TSCA							—	5.61×10^{-6}
SR-TSCA								—
Mean	18.09	17.56	17.47	20.42	18.00	16.46	16.65	18.06

TABLE 7.13: *Differences in respect of the mean time vehicles spent travelling under 10km/h obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	5.60×10^{-1}	3.26×10^{-1}	1.48×10^{-11}	4.80×10^{-3}	9.99×10^{-1}	1.10×10^{-2}	9.99×10^{-1}
O-TSCA(n)		—	3.99×10^{-4}	0.00×10^{-17}	1.81×10^{-1}	5.48×10^{-2}	2.48×10^{-6}	2.72×10^{-1}
Hybrid(n)			—	1.21×10^{-11}	2.56×10^{-7}	1.66×10^{-1}	7.30×10^{-1}	7.42×10^{-1}
Gersh				—	1.21×10^{-11}	1.10×10^{-12}	1.49×10^{-11}	1.28×10^{-11}
LH					—	2.72×10^{-5}	2.21×10^{-9}	1.54×10^{-3}
Fixed						—	1.74×10^{-3}	9.99×10^{-1}
VP-TSCA							—	7.19×10^{-2}
SR-TSCA								—
Mean	0.1739	0.1765	0.1707	0.1986	0.1796	0.1735	0.1684	0.1731

TABLE 7.14: *Differences in respect of the normalised mean time vehicles spent travelling under 10km/h obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a four-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

7.3.2 Algorithmic comparison in a grid network

The relative performances of the eight algorithms (the two new algorithms of §7 and the six existing algorithms of §5) are compared with one another in respect of the six PMIs of §4.1.4 within the context of a 3×4 grid of intersections under both lighter and heavier traffic conditions.

Simulation results obtained under lighter traffic conditions ($\lambda = 10$ vehicles/min)

The ANOVA column in Table 7.15 indicates that for all six PMIs there are statistical differences between the means returned by the algorithms at a 5% level of significance. Furthermore, the results of the Levene tests revealed that the variances of the mean samples are not statistically distinguishable for PMI 5; therefore the Fisher LSD *post hoc* test was employed in order to determine between which pairs of algorithmic output statistical differences between means are distinguishable. There was, however, a significant difference between the variance of the means for the remaining PMIs. Therefore, the Games-Howell *post hoc* test was performed for this purpose in respect of PMI 1, PMI 2, PMI 3, PMI 4 and PMI 6.

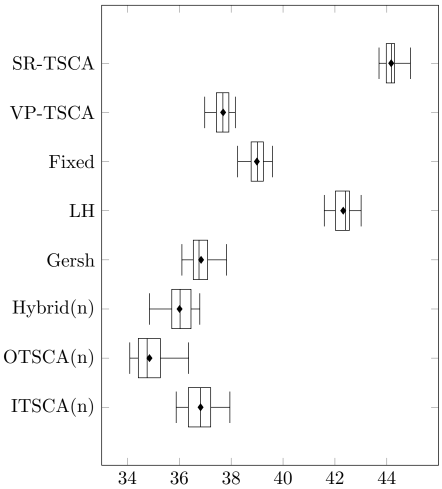
	Mean value								p-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA	ANOVA	Levene's Test
1	36.82	34.85	36.01	36.84	42.32	38.99	37.69	44.17	$<1\times10^{-17}$	2.98×10^{-4}
2	1.396	1.376	1.388	1.396	1.453	1.419	1.405	1.470	$<1\times10^{-17}$	4.48×10^{-5}
3	0.277	0.303	0.256	0.219	0.472	0.413	0.208	0.669	$<1\times10^{-17}$	1.41×10^{-3}
4	0.087	0.095	0.082	0.069	0.146	0.128	0.065	0.200	$<1\times10^{-17}$	2.34×10^{-2}
5	14.34	13.40	13.70	14.24	19.15	16.18	14.29	23.06	$<1\times10^{-17}$	8.82×10^{-2}
6	0.1017	0.0964	0.0978	0.1003	0.1308	0.1135	0.1015	0.1547	$<1\times10^{-17}$	1.15×10^{-2}

TABLE 7.15: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under lighter traffic conditions in a 3×4 grid of intersections. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

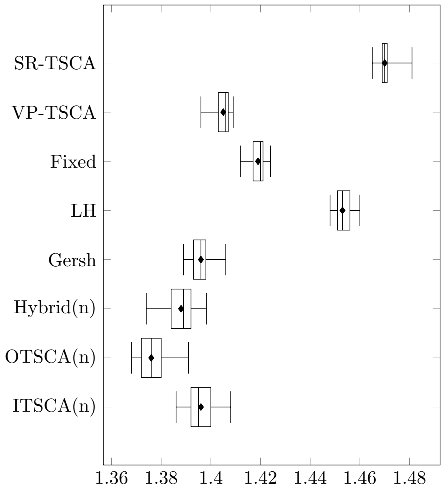
The VP-TSCA achieved a mean delay of 37.69 seconds and a normalised mean delay of 1.405 (see Tables 7.16 and 7.17), that is, an improvement of 15% and 4%, respectively, over the values achieved by the SR-TSCA, which was statistically the worst performing algorithm in respect of these two PMIs. The VP-TSCA ranked 5^{th} out of the eight algorithms in terms of both mean delay and normalised mean delay, outperforming only Fixed, LH and the SR-TSCA.

In terms of mean number of stops and normalised mean number of stops, the VP-TSCA returned favourable results, significantly outperforming all other algorithms at a 95% level of confidence, achieving values of 0.208 and 0.065, respectively (see Tables 7.18 and 7.19). This indicates that vehicles under the control of the VP-TSCA stop an average of 0.208 times throughout their journey through the road network and stop at 6.5% of the intersections they encounter, suggesting that effective coordination of green waves was achieved. The SR-TSCA, on the other hand, performed more than three times worse than the VP-TSCA, obtaining a mean number of stops value of 0.669 and a normalised mean number of stops value of 0.2. The SR-TSCA was again the worst performing algorithm with respect to these two PMIs as it failed to achieve good traffic flow coordination between consecutive intersections.

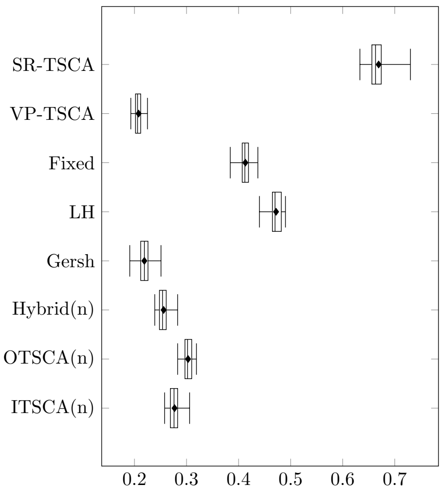
The SR-TSCA obtained particularly poor results in terms of the mean time spent travelling under 10km/h as well as normalised mean time spent travelling under 10km/h, returning values of 23.06 seconds and 0.1547, respectively (see Tables 7.20 and 7.21). The VP-TSCA achieved a relatively good result in terms of the mean time spent travelling under 10km/h, returning



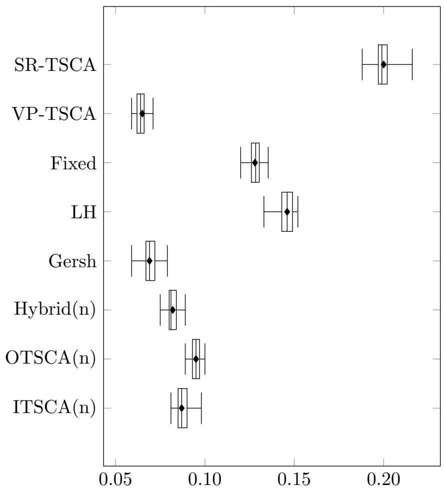
(a) *Mean delay*



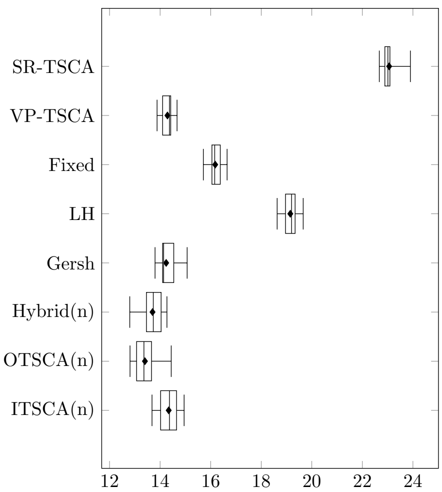
(b) *Normalised mean delay*



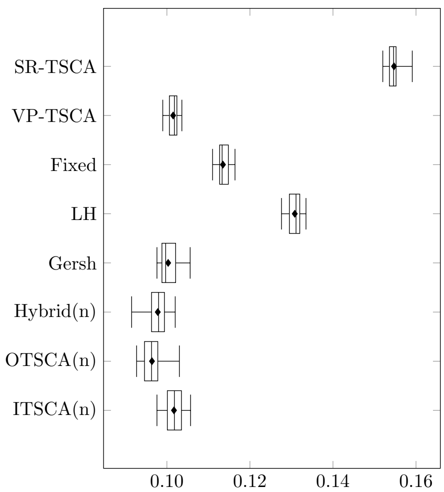
(c) *Mean number of stops*



(d) *Normalised mean number of stops*



(e) *Mean time spent travelling under 10km/h*



(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 7.5: *PMI results for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 within the context of a regular 3×4 grid of intersections under lighter traffic conditions.*

	p-values of the Games-Howell test: Mean delay							
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.45×10^{-11}	2.17×10^{-5}	9.99×10^{-1}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	1.44×10^{-7}	1.10×10^{-12}
O-TSCA(n)		—	1.21×10^{-9}	8.84×10^{-13}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	8.21×10^{-13}	1.02×10^{-12}
Hybrid(n)			—	5.61×10^{-7}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	3.37×10^{-13}	7.45×10^{-13}
Gersh				—	1.48×10^{-12}	7.50×10^{-12}	9.83×10^{-11}	$<1 \times 10^{-17}$
LH					—	1.37×10^{-11}	1.29×10^{-11}	1.17×10^{-11}
Fixed						—	8.53×10^{-12}	6.57×10^{-12}
VP-TSCA							—	1.47×10^{-11}
SR-TSCA								—
Mean	36.82	34.85	36.01	36.84	42.32	38.99	37.69	44.17

TABLE 7.16: *Differences in respect of the mean delay obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	p-values of the Games-Howell test: Normalised mean delay							
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.34×10^{-11}	9.39×10^{-5}	9.99×10^{-1}	4.80×10^{-13}	1.68×10^{-13}	7.23×10^{-8}	1.11×10^{-12}
O-TSCA(n)		—	1.15×10^{-9}	6.58×10^{-12}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	4.98×10^{-13}
Hybrid(n)			—	1.28×10^{-5}	6.00×10^{-13}	3.34×10^{-13}	9.59×10^{-13}	1.10×10^{-12}
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	3.57×10^{-11}	$<1 \times 10^{-17}$
LH					—	1.48×10^{-11}	1.47×10^{-11}	1.33×10^{-11}
Fixed						—	1.45×10^{-11}	1.27×10^{-11}
VP-TSCA							—	1.42×10^{-11}
SR-TSCA								—
Mean	1.396	1.376	1.388	1.396	1.453	1.419	1.405	1.470

TABLE 7.17: *Differences in respect of the normalised mean delay obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	8.38×10^{-12}	1.12×10^{-9}	1.47×10^{-11}	1.29×10^{-11}	1.46×10^{-11}	8.71×10^{-12}	1.02×10^{-12}
O-TSCA(n)		—	1.43×10^{-11}	5.61×10^{-12}	3.20×10^{-14}	4.51×10^{-12}	1.49×10^{-11}	2.34×10^{-13}
Hybrid(n)			—	9.63×10^{-12}	4.22×10^{-12}	8.69×10^{-12}	1.46×10^{-11}	6.14×10^{-13}
Gersh				—	1.39×10^{-11}	1.49×10^{-11}	9.45×10^{-4}	7.28×10^{-13}
LH					—	1.42×10^{-11}	1.03×10^{-12}	$<1 \times 10^{-17}$
Fixed						—	5.61×10^{-12}	5.36×10^{-13}
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	0.277	0.303	0.256	0.219	0.472	0.413	0.208	0.669

TABLE 7.18: *Differences in respect of the mean number of vehicle stops obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a* 3×4 *grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	8.51×10^{-11}	4.33×10^{-7}	1.43×10^{-11}	9.64×10^{-12}	1.47×10^{-11}	1.11×10^{-11}	$<1 \times 10^{-17}$
O-TSCA(n)		—	1.28×10^{-11}	3.50×10^{-12}	$<1 \times 10^{-17}$	5.12×10^{-12}	1.44×10^{-11}	1.01×10^{-12}
Hybrid(n)			—	1.09×10^{-11}	3.31×10^{-12}	1.20×10^{-11}	1.43×10^{-11}	7.05×10^{-13}
Gersh				—	1.27×10^{-11}	1.48×10^{-11}	4.76×10^{-5}	$<1 \times 10^{-17}$
LH					—	1.17×10^{-11}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Fixed						—	8.84×10^{-12}	$<1 \times 10^{-17}$
VP-TSCA							—	1.11×10^{-12}
SR-TSCA								—
Mean	0.087	0.095	0.082	0.069	0.146	0.128	0.065	0.200

TABLE 7.19: *Differences in respect of the normalised mean number of stops obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a* 3×4 *grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	$<1 \times 10^{-17}$	2.03×10^{-13}	2.04×10^{-1}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	5.16×10^{-1}	$<1 \times 10^{-17}$
O-TSCA(n)		—	2.51×10^{-4}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Hybrid(n)			—	4.08×10^{-10}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	1.09×10^{-11}	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	5.34×10^{-1}	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
Fixed						—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	14.34	13.40	13.70	14.24	19.15	16.18	14.29	23.06

TABLE 7.20: *Differences in respect of the mean time vehicles spent travelling under 10km/h obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a* 3×4 *grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	7.75×10^{-11}	1.30×10^{-6}	1.94×10^{-1}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	9.99×10^{-1}	$<1 \times 10^{-17}$
O-TSCA(n)		—	4.22×10^{-1}	2.26×10^{-7}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	7.05×10^{-12}	$<1 \times 10^{-17}$
Hybrid(n)			—	2.49×10^{-3}	$<1 \times 10^{-17}$	5.53×10^{-13}	3.43×10^{-7}	$<1 \times 10^{-17}$
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	1.50×10^{-1}	5.25×10^{-12}
LH					—	1.38×10^{-11}	1.23×10^{-11}	1.32×10^{-11}
Fixed						—	1.46×10^{-11}	9.35×10^{-12}
VP-TSCA							—	6.44×10^{-12}
SR-TSCA								—
Mean	0.1017	0.0964	0.0978	0.1003	0.1308	0.1135	0.1015	0.1547

TABLE 7.21: *Differences in respect of the normalised mean time vehicles spent travelling under 10km/h obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under lighter traffic conditions within the context of a* 3×4 *grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

a value of 14.29 seconds, a 38% improvement over that achieved by the SR-TSCA. The VP-TSCA obtained a value of 0.1015 for normalised mean time spent travelling under 10km/h, outperforming Fixed, LH and SR-TSCA, but not differing significantly from the I-TSCA(n).

It is clear that the SR-TSCA does not perform well under lighter traffic conditions — it was consistently the worst performing algorithm over all six of the PMIs as is clear from the box plots in Figure 7.5. While the VP-TSCA was very effective in respect of the PMIs related to the mean number of stops, its performance was relatively average for the other four PMIs, outperforming the worst algorithms, but not yielding the best results.

Simulation results obtained under heavier traffic conditions ($\lambda = 20$ vehicles/min)

Once again the ANOVA column in Table 7.22 indicates that for all six PMIs there are statistical differences between the means of the algorithms at a 5% level of significance. Furthermore, the results of the Levene tests revealed that the variances of the mean samples were statistically different for all six of the PMIs. Therefore the Games-Howell test was performed for all six PMIs to determine between which pairs of algorithmic output statistical differences between means are distinguishable.

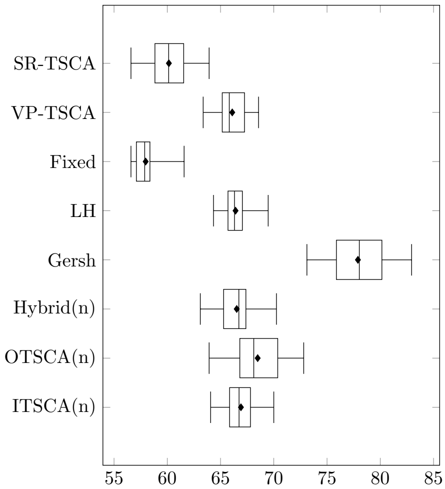
	Mean value								*p*-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA	ANOVA	Levene's Test
1	66.92	68.49	66.52	77.90	66.41	57.97	66.10	60.15	$<1 \times 10^{-17}$	1.03×10^{-5}
2	1.716	1.731	1.714	1.834	1.711	1.630	1.707	1.651	$<1 \times 10^{-17}$	4.61×10^{-6}
3	1.661	1.762	1.762	3.359	1.637	1.745	1.862	1.822	$<1 \times 10^{-17}$	1.40×10^{-6}
4	0.522	0.547	0.557	1.038	0.516	0.575	0.584	0.580	$<1 \times 10^{-17}$	3.47×10^{-7}
5	35.38	36.67	34.76	44.64	35.37	28.26	34.26	32.37	$<1 \times 10^{-17}$	4.76×10^{-6}
6	0.2038	0.2113	0.2015	0.2359	0.2065	0.1682	0.1975	0.1922	$<1 \times 10^{-17}$	3.89×10^{-6}

TABLE 7.22: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under heavier traffic conditions in a 3×4 grid of intersections. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

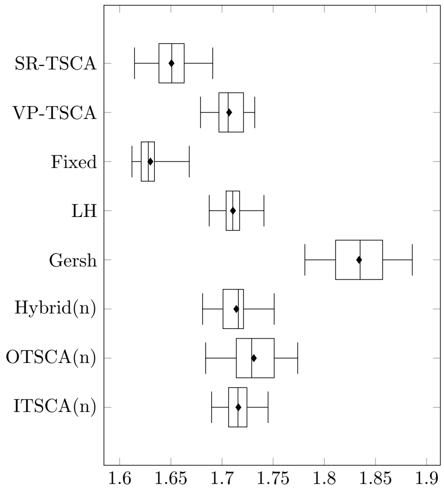
The SR-TSCA achieved the second best results in terms of mean delay (after Fixed), significantly outperforming the VP-TSCA, LH, Hybrid(n), the I-TSCA(n), the O-TSCA(n) and Gersh by margins of 9%, 9%, 10%, 10%, 12% and 23%, respectively (see Table 7.23). The VP-TSCA also performed relatively well, outperforming the O-TSCA(n) and Gersh, while not differing statistically from the I-TSCA(n), Hybrid(n) and LH at a 5% level of significance. The results obtained for the normalised mean delay are similar for all algorithms, with the SR-TSCA outperforming all other algorithms except Fixed, and returning a value of 1.651, while the VP-TSCA significantly outperformed the O-TSCA(n) and Gersh with a value of 1.707 (see Table 7.24).

Interestingly, the VP-TSCA did not perform as well under heavy traffic conditions in terms of the mean number of stops or the normalised mean number of stops. It improved only upon Gersh, obtaining values of 1.862 and 0.584, respectively (see Tables 7.25 and 7.26). The SR-TSCA achieved marginally better results, and was only outperformed by the I-TSCA(n) and LH in terms of the mean number of stops, by 10% and 11%, respectively. Furthermore the SR-TSCA was outperformed by the I-TSCA(n) and LH in terms of normalised mean number of stops, by margins of 11% and 12%, respectively.

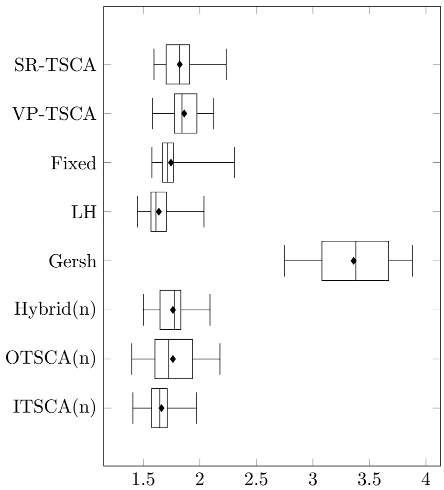
The SR-TSCA achieved the second smallest value for the mean time spent travelling under 10km/h (after Fixed) as well as for the normalised mean time spent travelling under 10km/h,



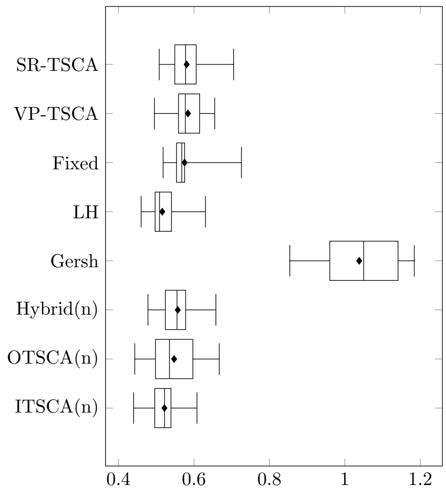
(a) *Mean delay*



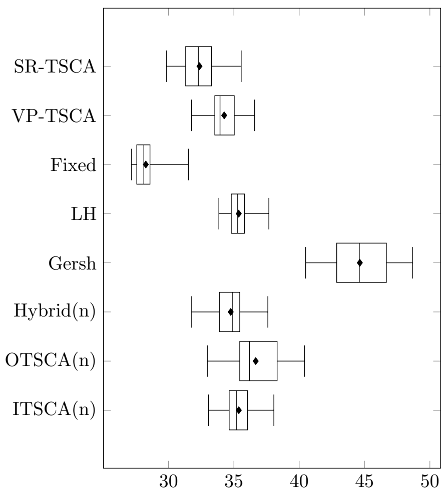
(b) *Normalised mean delay*



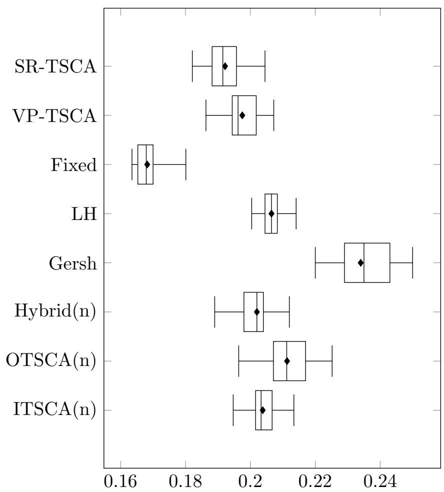
(c) *Mean number of stops*



(d) *Normalised mean number of stops*



(e) *Mean time spent travelling under 10km/h*



(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 7.6: *PMI results for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 within the context of a regular 3×4 grid of intersections under heavier traffic conditions.*

p-values of the Games-Howell test: Mean delay								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	4.21×10^{-12}	9.75×10^{-1}	1.10×10^{-12}	7.51×10^{-1}	$<1 \times 10^{-17}$	3.22×10^{-1}	1.87×10^{-12}
O-TSCA(n)		—	8.74×10^{-3}	8.57×10^{-12}	1.24×10^{-3}	4.67×10^{-12}	3.00×10^{-4}	2.53×10^{-12}
Hybrid(n)			—	$<1 \times 10^{-17}$	9.99×10^{-1}	1.39×10^{-11}	9.63×10^{-1}	1.45×10^{-11}
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	9.86×10^{-13}	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	9.77×10^{-1}
Fixed						—	2.12×10^{-12}	4.71×10^{-3}
VP-TSCA							—	4.27×10^{-12}
SR-TSCA								—
Mean	66.92	68.49	66.52	77.90	66.41	57.97	66.10	60.15

Table 7.23: *Differences in respect of the mean delay obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean delay								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	5.63×10^{-2}	9.98×10^{-1}	9.01×10^{-13}	6.94×10^{-1}	$<1 \times 10^{-17}$	2.05×10^{-1}	$<1 \times 10^{-17}$
O-TSCA(n)		—	2.94×10^{-2}	4.64×10^{-12}	1.62×10^{-3}	7.71×10^{-12}	2.67×10^{-4}	4.15×10^{-12}
Hybrid(n)			—	$<1 \times 10^{-17}$	9.95×10^{-1}	1.39×10^{-11}	7.59×10^{-1}	1.48×10^{-11}
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	1.02×10^{-12}	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$	9.46×10^{-1}	$<1 \times 10^{-17}$
Fixed						—	2.71×10^{-12}	1.62×10^{-2}
VP-TSCA							—	6.35×10^{-12}
SR-TSCA								—
Mean	1.716	1.731	1.714	1.834	1.711	1.630	1.707	1.651

Table 7.24: *Differences in respect of the normalised mean delay obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	3.49×10^{-1}	1.15×10^{-1}	$<1 \times 10^{-17}$	9.95×10^{-1}	3.80×10^{-1}	6.27×10^{-6}	5.21×10^{-4}
O-TSCA(n)		—	9.99×10^{-1}	$<1 \times 10^{-17}$	1.12×10^{-1}	9.86×10^{-1}	3.60×10^{-1}	8.84×10^{-1}
Hybrid(n)			—	5.89×10^{-13}	1.54×10^{-2}	9.82×10^{-1}	1.36×10^{-1}	7.37×10^{-1}
Gersh				—	$<1 \times 10^{-17}$	2.95×10^{-12}	$<1 \times 10^{-17}$	1.19×10^{-13}
LH					—	2.26×10^{-1}	1.80×10^{-7}	2.29×10^{-5}
Fixed						—	9.99×10^{-1}	9.99×10^{-1}
VP-TSCA							—	9.48×10^{-1}
SR-TSCA								—
Mean	1.661	1.762	1.762	3.359	1.637	1.745	1.862	1.822

TABLE 7.25: *Differences in respect of the mean number of vehicle stops obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	5.06×10^{-1}	3.35×10^{-2}	$<1 \times 10^{-17}$	9.98×10^{-1}	2.46×10^{-2}	1.79×10^{-6}	1.01×10^{-5}
O-TSCA(n)		—	9.95×10^{-1}	$<1 \times 10^{-17}$	2.32×10^{-1}	3.45×10^{-1}	1.18×10^{-1}	2.17×10^{-1}
Hybrid(n)			—	7.97×10^{-13}	5.15×10^{-3}	5.47×10^{-1}	2.52×10^{-1}	4.47×10^{-1}
Gersh				—	$<1 \times 10^{-17}$	6.53×10^{-12}	$<1 \times 10^{-17}$	1.07×10^{-13}
LH					—	1.24×10^{-2}	8.10×10^{-8}	5.58×10^{-7}
Fixed						—	9.93×10^{-1}	9.80×10^{-1}
VP-TSCA							—	9.99×10^{-1}
SR-TSCA								—
Mean	0.522	0.547	0.557	1.038	0.516	0.575	0.584	0.580

TABLE 7.26: *Differences in respect of the normalised mean number of stops obtained the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	3.83×10^{-2}	5.27×10^{-1}	8.35×10^{-13}	9.99×10^{-1}	$<1 \times 10^{-17}$	7.34×10^{-3}	1.28×10^{-11}
O-TSCA(n)		—	8.83×10^{-4}	5.81×10^{-12}	2.59×10^{-2}	1.25×10^{-11}	7.24×10^{-6}	$<1 \times 10^{-17}$
Hybrid(n)			—	$<1 \times 10^{-17}$	4.31×10^{-1}	6.58×10^{-12}	8.08×10^{-1}	1.88×10^{-7}
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	1.09×10^{-12}	$<1 \times 10^{-17}$
LH					—	1.14×10^{-12}	2.43×10^{-3}	$<1 \times 10^{-17}$
Fixed						—	$<1 \times 10^{-17}$	1.08×10^{-11}
VP-TSCA							—	8.48×10^{-6}
SR-TSCA								—
Mean	35.38	36.67	34.76	44.64	35.37	28.26	34.26	32.37

TABLE 7.27: *Differences in respect of the mean time vehicles spent travelling under 10km/h obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a 3 × 4 grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.69×10^{-4}	5.99×10^{-1}	8.87×10^{-13}	9.21×10^{-2}	5.00×10^{-12}	2.58×10^{-5}	1.19×10^{-11}
O-TSCA(n)		—	4.79×10^{-6}	8.82×10^{-12}	2.97×10^{-2}	$<1 \times 10^{-17}$	1.93×10^{-10}	9.91×10^{-13}
Hybrid(n)			—	$<1 \times 10^{-17}$	2.44×10^{-3}	1.17×10^{-11}	7.54×10^{-2}	4.28×10^{-7}
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$	1.01×10^{-9}	$<1 \times 10^{-17}$
Fixed						—	1.49×10^{-11}	1.32×10^{-11}
VP-TSCA							—	5.04×10^{-3}
SR-TSCA								—
Mean	0.2038	0.2113	0.2015	0.2359	0.2065	0.1682	0.1975	0.1922

TABLE 7.28: *Differences in respect of the normalised mean time vehicles spent travelling under 10km/h obtained for the Fixed algorithm and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under heavier traffic conditions within the context of a 3 × 4 grid of intersections. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

obtaining values of 32.37 seconds and 0.1922, respectively (see Tables 7.27 and 7.28). The VP-TSCA was only outperformed by Fixed and the SR-TSCA in respect of mean time spent travelling under 10km/h, although it does not differ statistically from Hybrid(n) at a 5% level of significance. The VP-TSCA is statistically the third best performing algorithm in terms of normalised mean time spent travelling under 10km/h, and is only outperformed by the SR-TSCA and Fixed, by a margins of 3% and 16%, respectively, at a 95% level of confidence.

The SR-TSCA once again performed effectively under heavier traffic conditions which is clear from the box plots in Figure 7.6, obtaining good values for four of the six PMIs, while returning relatively average results for the mean number of stops and the normalised mean number of stops. The VP-TSCA obtained average results relative to the other six algorithms, never achieving the best or worst PMI value.

7.4 Chapter summary

This chapter opened in §7.1 with a description of a new self-organising algorithm that partitions vehicles into clusters. This was followed by the proposal of another new self-organising algorithm in §7.2 that deals with vehicle saturation levels of competing traffic flows. The relative performances of these two new algorithms were compared with one another as well as with those of LH, Gersh and the algorithms of §5.3 in the context of a four-intersection corridor and a regular 3×4 grid of intersections under lighter and heavier traffic conditions in §7.3. The VP-TSCA and Hybrid(n) were found to be the most successful algorithms in the context of a corridor road network under lighter traffic conditions, while Fixed and VP-TSCA yielded the best results for a corridor under heavier traffic conditions. The VP-TSCA and Gersh furthermore obtained the best results for the grid road network under lighter traffic conditions, while Fixed returned the most favourable results under heavier traffic conditions in a grid network, followed by the SR-TSCA.

CHAPTER 8

Simulation results

Contents

This chapter consists of the results of four different experiments that take place in the context of a number of different road network topologies under varying traffic conditions. The first two experiments take place in a grid network, considering a road closure (in §8.1.1) in the one, and a varying arrival rate (in §8.1.2) in the other. The final two experiments take place in a road corridor. The first of these involve the scaling of a corridor in §8.2.1, while the second involves a six-intersection corridor in which distances between neighbouring intersections are varied in §8.2.2. The chapter closes in §8.3 with a summary of the results.

8.1 A 3×4 grid road network topology

Two different scenarios within a 3×4 grid road network are considered in this section. The first scenario includes a constant arrival rate and one missing road link in the centre of the road network, representing a road closure which is a relatively common occurrence in reality. This scenario is considered in order to determine how well the algorithms are able to respond to the increased congestion occurring on the roads that remain open. The second scenario involves a variable arrival rate, with all road sections open. This scenario mimics a sudden increased arrival rate of vehicles to a particular point in the road network, symbolising a large event taking place (such as a music concert or sports event) at a specific time, and is aimed at ascertaining how well the algorithms are capable of handling a sudden increase in vehicles within a specific area of the network.

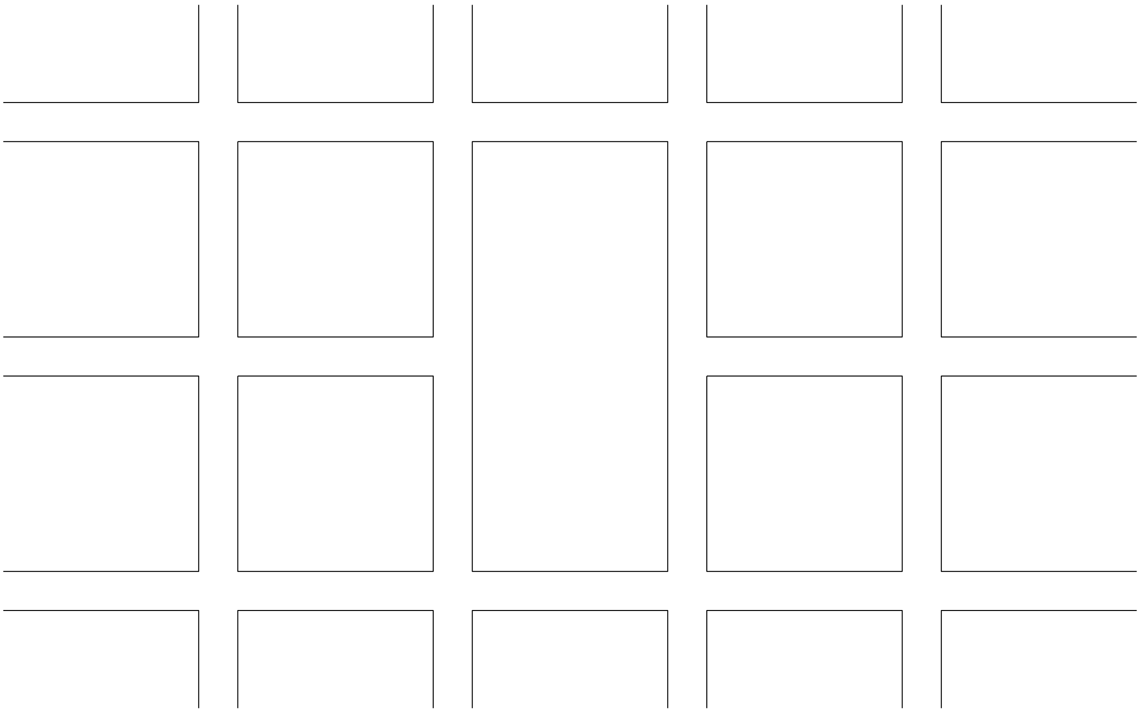


FIGURE 8.1: *A* 3×4 *grid of intersections with a closed horizontal road in the centre of the road network.*

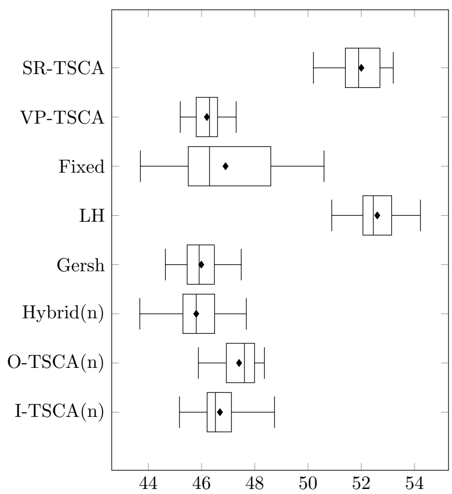
8.1.1 A road closure

The scenario considered in this section involves a 3×4 grid network of equally spaced intersections similar to the one described in §6.3. A section of road is, however, removed from this road network in order to mimic a situation where a road is closed due to road works or an accident, for example. The specific road that is closed is the horizontal road in the centre of the network shown in Figure 8.1. A light traffic flow of $\lambda = 10$ vehicles per minute is the arrival rate implemented in the model described in Chapter 4. The reason for choosing a relatively low traffic flow is due to the fact that the road closure causes a significant increase in traffic congestion, particularly around the remaining road segments above and below the one that is closed. This scenario was implemented in the simulation software environment in order to test how the performance of the algorithms differ from one another when certain areas of the road network become more saturated.

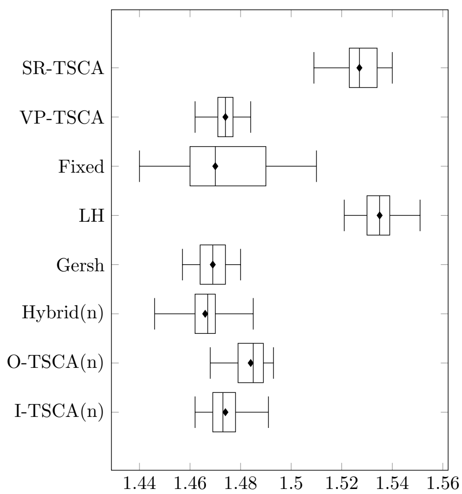
Simulation results

The ANOVA column in Table 8.1 indicates that for all six PMIs there are statistical differences between the means of the algorithms at a 5% level of significance. Furthermore, the results of the Levene tests revealed that the variances of the mean samples are statistically distinguishable for all six PMIs; therefore the Games-Howell *post hoc* test was employed in order to determine between which pairs of algorithmic output statistical differences between means are distinguishable.

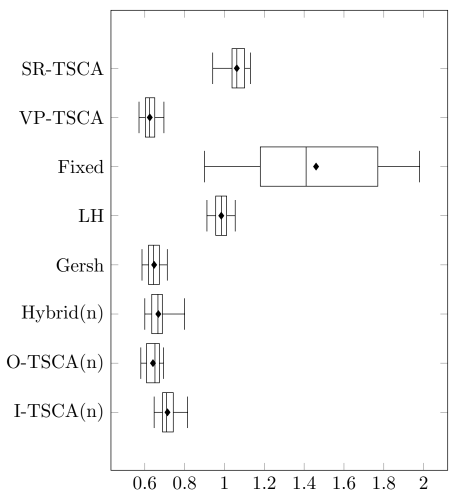
Hybrid(n) and Gersh obtained the most favourable results in terms of mean delay time (see Figure 8.2(a) and Table 8.2), achieving values of 45.80 seconds and 45.99 seconds, respectively. These algorithms did not differ significantly from each other, but they both statistically out-



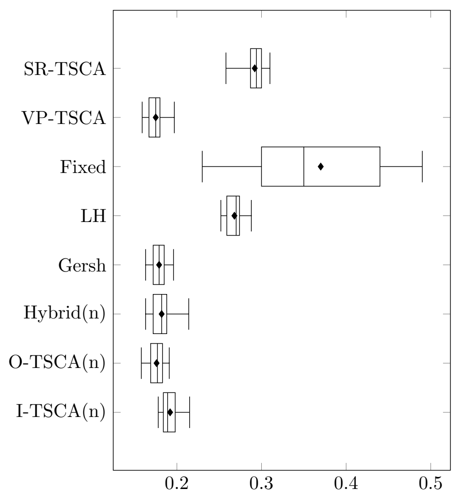
(a) *Mean delay time*



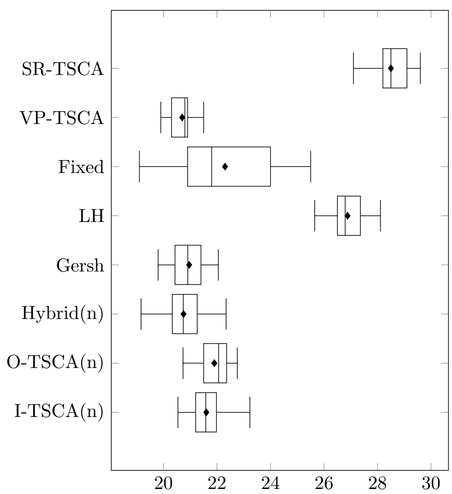
(b) *Normalised mean delay time*



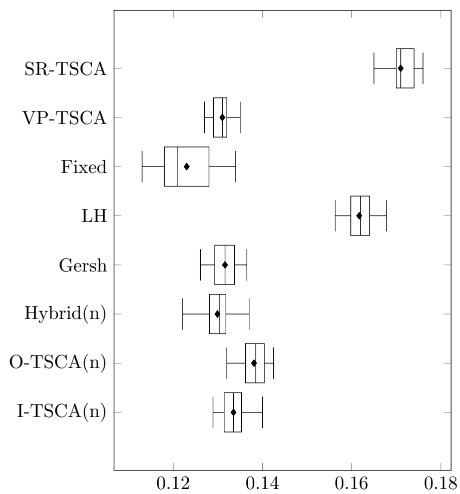
(c) *Mean number of stops*



(d) *Normalised mean number of stops*



(e) *Mean time spent under 10km/h*



(f) *Normalised mean time spent under 10km/h*

FIGURE 8.2: *PMI results for Fixed and the seven self-organising algorithms in a grid road network containing a closed road.*

	Mean value								p-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA	ANOVA	Levene's Test
1	46.69	47.41	45.80	45.99	52.60	46.89	46.25	52.01	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
2	1.474	1.484	1.467	1.469	1.535	1.472	1.474	1.527	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
3	0.714	0.641	0.668	0.647	0.984	1.463	0.625	1.062	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
4	0.192	0.176	0.182	0.179	0.268	0.369	0.175	0.292	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
5	21.60	21.91	20.75	20.96	26.88	22.26	20.68	28.53	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
6	0.1335	0.1381	0.1299	0.1316	0.1617	0.1228	0.1309	0.1714	$<1 \times 10^{-17}$	2.86×10^{-11}

TABLE 8.1: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests in the context of a 3×4 grid of intersections containing a closed road. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

performed the next best performing algorithms in this regard, the VP-TSCA and Fixed, which obtained mean values of 46.25 seconds and 46.89 seconds, respectively. These differences are not practically significant, as they differ by approximately one second from the values obtained by Hybrid(n) and Gersh. The O-TSCA(n) performed significantly worse than Hybrid(n), Gersh and the I-TSCA(n). In terms of the normalised mean delay time shown in Figure 8.2(b) and Table 8.3, Hybrid(n) achieved the best results as it was not outperformed by any other algorithm, while it outperformed the largest number of algorithms at a 5% level of significance. Gersh and Fixed obtained the next best results, obtaining values of 1.469 and 1.472, respectively. The O-TSCA(n) was outperformed by the previously mentioned algorithms as well as by the I-TSCA(n), but it outperformed both LH and the SR-TSCA significantly, achieving a value of 1.484, while LH and the SR-TSCA achieved values of 1.535 and 1.527, respectively.

The VP-TSCA obtained the best results in respect of the mean number of stops (see Figure 8.2(c) and Table 8.4), followed by the O-TSCA(n) and Gersh which did not differ significantly from one another. Fixed obtained very poor results in respect of this PMI returning a result of 1.463, implying that vehicles are required to stop approximately twice as many times when under the control of Fixed, in comparison to the other algorithms. The VP-TSCA, the O-TSCA(n), Hybrid(n) and Gersh all returned PMI results for normalised mean number of stops (see Figure 8.2(d) and Table 8.5) that were not distinguishable from one another at a 5% level of significance. These algorithms did, however, outperform the other four algorithms at a 5% level of significance, making them the best performing algorithms in respect of this particular PMI. Fixed was again the worst performing algorithm with a mean number of stops value of 1.463 and a normalised equivalent of 0.369, indicating that vehicles come to a complete stop at almost 37% of the intersections they encounter, which is substantially larger than for the other seven algorithms.

In respect of the mean time spent by vehicles travelling under 10km/h, Hybrid(n), the VP-TSCA and Gersh performed the best statistically (achieving values of 20.75 seconds, 20.68 seconds and 20.96 seconds, respectively) in comparison with the other algorithms (see Figure 8.2(e) and Table 8.6). They marginally outperformed the I-TSCA(n), the O-TSCA(n) and Fixed, none of which performed significantly differently from one another. Fixed achieved the best algorithmic performance in respect of the normalised mean time spent travelling under 10km/h, obtaining a value of 0.1228 which indicates vehicles spend approximately 12% of their time in the road network travelling under 10km/h. The next best algorithms in this respect were Hybrid(n), the VP-TSCA, Gersh and the I-TSCA(n) which did not perform significantly differently from one another at a 5% level of significance (see Figure 8.2(f) and Table 8.7).

	p-values of the Games-Howell test: Mean delay time							
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.00×10^{-2}	2.13×10^{-3}	1.61×10^{-2}	1.49×10^{-11}	9.99×10^{-1}	2.27×10^{-1}	1.42×10^{-11}
O-TSCA(n)		—	2.22×10^{-9}	6.89×10^{-9}	1.27×10^{-11}	8.82×10^{-1}	1.25×10^{-7}	9.59×10^{-12}
Hybrid(n)			—	9.79×10^{-1}	1.39×10^{-11}	1.39×10^{-1}	2.74×10^{-1}	1.49×10^{-11}
Gersh				—	1.35×10^{-11}	3.18×10^{-1}	8.13×10^{-1}	1.09×10^{-11}
LH					—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	1.11×10^{-1}
Fixed						—	6.91×10^{-1}	5.27×10^{-14}
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	46.69	47.41	45.80	45.99	52.60	46.89	46.25	52.01

TABLE 8.2: *Differences in respect of the mean delay obtained for each of the algorithms in the context of a 3 × 4 grid of intersections with a closed road. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	p-values of the Games-Howell test: Normalised mean delay time							
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	4.21×10^{-5}	8.06×10^{-3}	1.03×10^{-1}	1.44×10^{-11}	9.99×10^{-1}	9.99×10^{-1}	1.39×10^{-11}
O-TSCA(n)		—	7.73×10^{-11}	1.88×10^{-10}	1.48×10^{-11}	5.90×10^{-2}	3.53×10^{-6}	1.28×10^{-11}
Hybrid(n)			—	8.75×10^{-1}	9.80×10^{-12}	7.37×10^{-1}	2.61×10^{-3}	1.46×10^{-11}
Gersh				—	1.31×10^{-11}	9.76×10^{-1}	3.87×10^{-2}	5.58×10^{-12}
LH					—	$<1 \times 10^{-17}$	4.90×10^{-12}	5.57×10^{-3}
Fixed						—	9.99×10^{-1}	8.43×10^{-14}
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	1.474	1.484	1.467	1.469	1.535	1.472	1.474	1.527

TABLE 8.3: *Differences in respect of the normalised mean delay obtained for each of the algorithms in the context of a 3 × 4 grid of intersections with a closed road. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	4.12×10^{-9}	1.37×10^{-3}	9.54×10^{-8}	1.39×10^{-11}	8.40×10^{-12}	1.25×10^{-11}	1.08×10^{-12}
O-TSCA(n)		—	1.68×10^{-1}	9.98×10^{-1}	1.47×10^{-11}	8.74×10^{-13}	5.42×10^{-1}	$<1 \times 10^{-17}$
Hybrid(n)			—	5.00×10^{-1}	5.77×10^{-12}	1.79×10^{-12}	1.52×10^{-3}	1.14×10^{-11}
Gersh				—	1.48×10^{-11}	1.03×10^{-12}	1.96×10^{-1}	$<1 \times 10^{-17}$
LH					—	2.36×10^{-7}	1.34×10^{-11}	1.44×10^{-7}
Fixed						—	5.42×10^{-13}	6.80×10^{-6}
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	0.714	0.641	0.668	0.647	0.984	1.463	0.625	1.062

TABLE 8.4: *Differences in respect of the mean number of stops obtained for each of the algorithms in the context of a* 3×4 *grid of intersections with a closed road. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	2.75×10^{-7}	9.24×10^{-3}	4.30×10^{-5}	1.49×10^{-11}	2.64×10^{-11}	5.95×10^{-8}	$<1 \times 10^{-17}$
O-TSCA(n)		—	4.53×10^{-1}	9.14×10^{-1}	1.46×10^{-11}	3.25×10^{-12}	9.99×10^{-1}	$<1 \times 10^{-17}$
Hybrid(n)			—	9.79×10^{-1}	4.15×10^{-12}	5.84×10^{-12}	2.19×10^{-1}	1.24×10^{-11}
Gersh				—	1.49×10^{-11}	4.68×10^{-12}	6.55×10^{-1}	$<1 \times 10^{-17}$
LH					—	4.83×10^{-6}	1.49×10^{-11}	9.04×10^{-10}
Fixed						—	2.70×10^{-12}	4.19×10^{-4}
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	0.192	0.176	0.182	0.179	0.268	0.369	0.175	0.292

TABLE 8.5: *Differences in respect of the normalised mean number of stops obtained for each of the algorithms in the context of a* 3×4 *grid of intersections with a closed road. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean time spent under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	4.59×10^{-1}	9.64×10^{-5}	2.57×10^{-3}	1.46×10^{-11}	6.00×10^{-1}	5.64×10^{-7}	1.35×10^{-11}
O-TSCA(n)		—	2.40×10^{-8}	4.60×10^{-7}	1.36×10^{-11}	9.72×10^{-1}	1.55×10^{-11}	6.33×10^{-12}
Hybrid(n)			—	9.01×10^{-1}	1.25×10^{-11}	4.37×10^{-3}	9.99×10^{-1}	1.49×10^{-11}
Gersh				—	1.46×10^{-11}	1.83×10^{-2}	4.33×10^{-1}	9.79×10^{-12}
LH					—	1.50×10^{-13}	6.57×10^{-13}	1.24×10^{-11}
Fixed						—	1.99×10^{-3}	$<1 \times 10^{-17}$
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	21.60	21.91	20.75	20.96	26.88	22.26	20.68	28.53

TABLE 8.6: *Differences in respect of the mean time spent under 10km/h obtained for each of the algorithms in the context of a 3 × 4 grid of intersections with a closed road. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean time spent under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	3.53×10^{-7}	6.42×10^{-4}	1.40×10^{-1}	1.37×10^{-11}	1.26×10^{-9}	4.62×10^{-3}	1.48×10^{-11}
O-TSCA(n)		—	8.80×10^{-12}	2.00×10^{-11}	1.46×10^{-11}	5.30×10^{-13}	4.09×10^{-12}	1.41×10^{-11}
Hybrid(n)			—	4.03×10^{-1}	5.96×10^{-12}	1.85×10^{-5}	8.48×10^{-1}	1.24×10^{-11}
Gersh				—	1.46×10^{-11}	1.76×10^{-7}	9.69×10^{-1}	1.41×10^{-11}
LH					—	2.29×10^{-13}	6.90×10^{-12}	1.29×10^{-11}
Fixed						—	8.19×10^{-7}	9.50×10^{-13}
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	0.1335	0.1381	0.1299	0.1316	0.1617	0.1228	0.1309	0.1714

TABLE 8.7: *Differences in respect of the normalised mean time spent under 10km/h obtained for each of the algorithms in the context of a 3 × 4 grid of intersections with a closed road. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

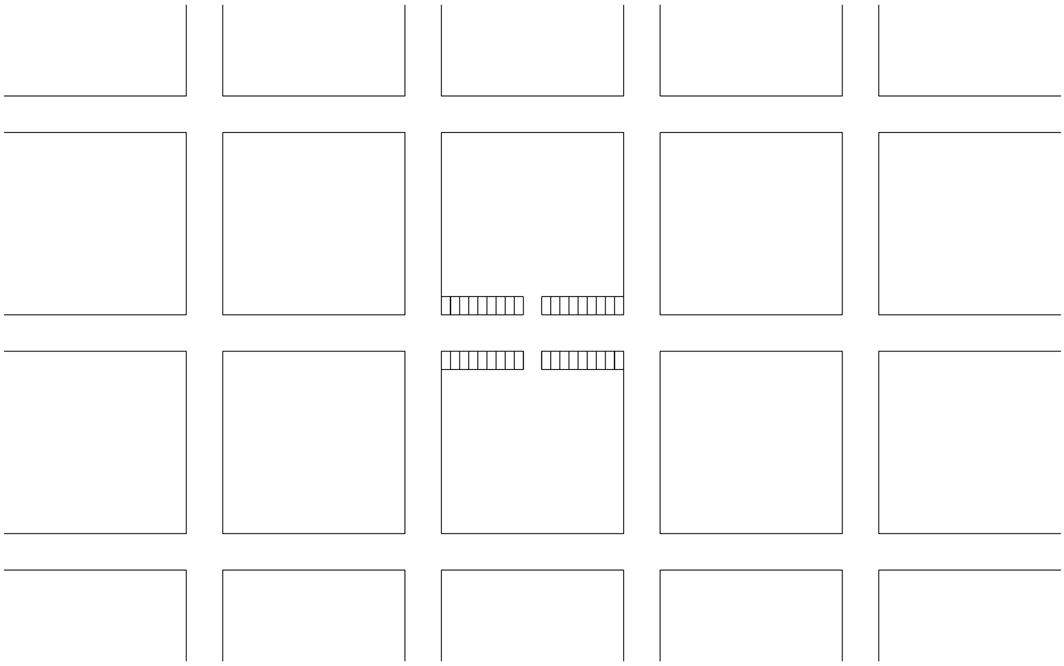


FIGURE 8.3: *A 3×4 grid of intersections with an increased arrival rate to a particular point in the centre of the road network. The rectangular blocks on either side of the road represent parking spaces in which vehicles travelling to the event may park.*

Although traffic conditions become heavier than those experienced under light traffic conditions (due to the congestion surrounding the remaining roads in the network), LH still performed poorly, which was an unexpected result as it was found to perform relatively well under heavier traffic conditions in §6.3. The reason for this is the very short green times that LH allocated when there is a much larger demand in the one direction along a road section than along the other. In this case, LH favoured the direction that exhibited the larger demand, so much so that the other direction suffered, receiving green times as short as one second. This was extremely ineffective as not only can a single vehicle not make it through the intersection during such a short interval, but the setup times involved are wasted when this occurs. This shortcoming may be easily rectified, by setting a minimum green time. Gersh, Hybrid(n) and the VP-TSCA were statistically the best performing algorithms overall. While Fixed performed very poorly in respect of the two PMIs related to the mean number of vehicle stops, it did perform relatively well in respect of the delay and slow speed PMIs.

8.1.2 A varying arrival rate

The arrival rate of vehicles is varied in this section throughout the duration of an unfolding scenario involving a 3×4 grid of intersections. Initially the arrival rate was set to $\lambda = 12$ vehicles per minute, with vehicles travelling to destinations that lie outside the road network as was described in §4.3.2. Once the simulation had warmed up, the arrival rate was increased to $\lambda = 15$ vehicles per minute, which resulted in a total of 42 additional vehicles entering the network per minute, on average. These vehicles were all directed to a point in the middle of the road network indicated in Figure 8.3. This scenario was designed to test how well the algorithms recover from a large influx of vehicles to a certain location in the road network. The scenario may represent the commencement of a large event such as a sports game or concert.

In this experiment, the simulation was first warmed up with an arrival rate of $\lambda = 12$ for 1 800 seconds (30 minutes). The arrival rate was then increased to $\lambda = 15$ for another 1 800 seconds, after which was decreased back to $\lambda = 12$ for a final 2 400 seconds (40 minutes). The six PMIs for each algorithm were cumulatively averaged and recorded each simulated minute, starting from 1 860 seconds (*i.e.* one minute after the arrival rate had been increased to $\lambda = 15$; therefore the PMIs were recorded over 70 simulated minutes in total). These arrival rates and time intervals were chosen specifically to ensure a significant increase in traffic flow, particularly around the area in the network where the vehicles were arriving — large enough to allow a slight spillback into the road, but small enough to ensure that the queues did not back up into intersections.

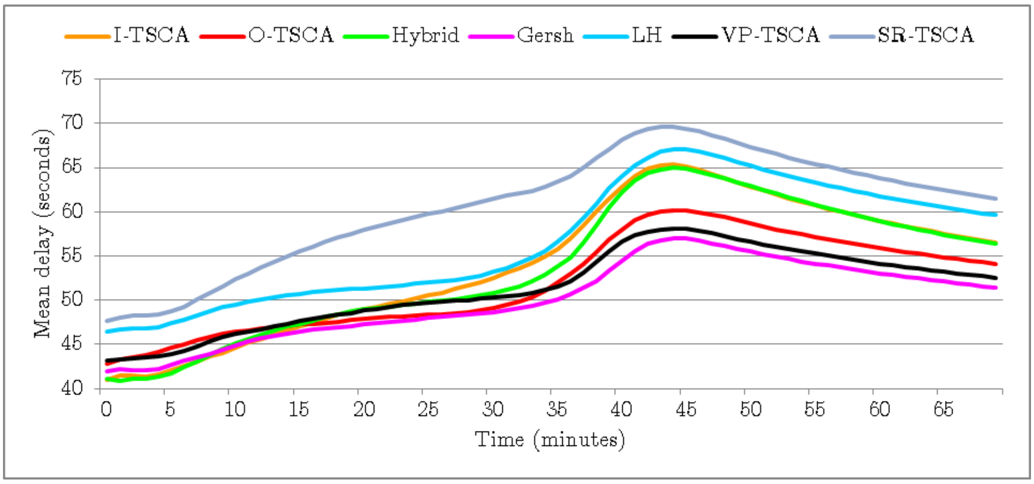
The results returned by the eight algorithms are not presented as box plots as before, because the PMIs change considerably throughout the course of a simulation run. The end mean was therefore not a good indication of how the algorithms performed. The results are rather presented in the form of graphs, one for each PMI, as they change over time. Initially, the PMIs were recorded for all vehicles in the road network, but it was difficult to distinguish between the delay experienced by vehicles arriving at the event and vehicles that were just travelling through the network. It is important to differentiate between these two kinds of vehicles as the more delayed vehicles tended to be those in the event location as they park, more so than those travelling through the road network. The PMIs of vehicles travelling to the event were therefore excluded.

There were parking areas on both sides of the road, so that vehicles were not required to turn right in front of oncoming traffic and hold up other vehicles travelling behind them. It is assumed that there are enough parking spaces for the vehicles arriving at the event location and so the parking spaces were modelled as sinks into which vehicles disappear.

Simulation results

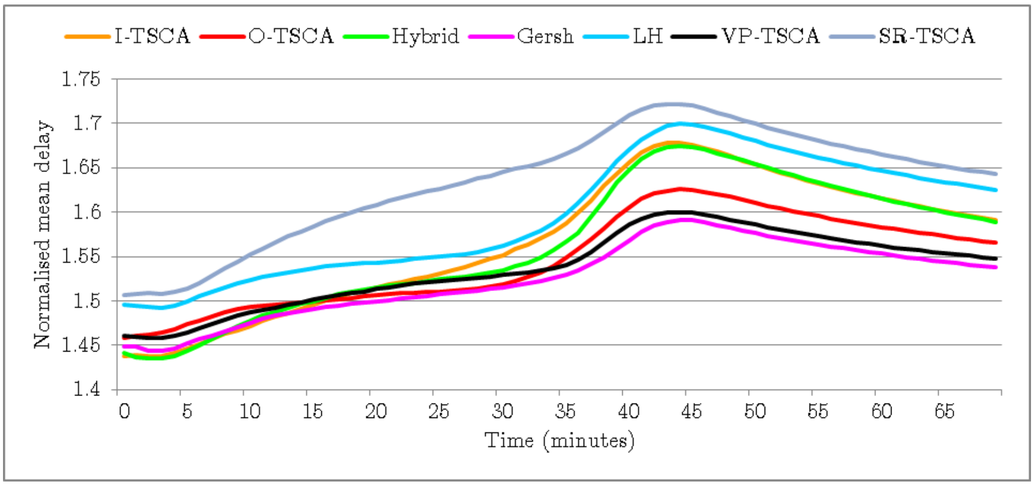
In this scenario, the PMIs achieved by the algorithms were recorded each simulated minute, as opposed to an average taken over the entire simulation run as was done previously. A single measure that is representative of the curves in Figures 8.4–8.9 nevertheless had to be chosen in order to allow for a comparison of the algorithms with one another (*i.e.* an instant in time or the area under the curve) [39]. This value was chosen as the 45^{th} simulated minute during which the largest PMI value occurs, on average. The significant differences between the PMI values returned by the algorithms are shown in Tables 8.9–8.14. Fixed yielded very poor results and was therefore excluded from the graphs so that the relative performances of the other algorithms could be seen more clearly in the graphs.

All six of the graphs in Figures 8.4–8.9 follow a similar trend over the course of the 70 minute simulated period. During the first five minutes, the PMI values remained relatively constant, as the increase of vehicles entering the network had little effect on the vehicles exiting the network. From the 5-minute mark until approximately the 30^{th} simulated minute there was a constant increase in the PMI values as a result of the additional influx of vehicles arriving in the network, although this value was less clear in the mean number of stops and normalised mean number of stops. The reason for this is that the increase in traffic flow caused vehicles to travel slower due to the additional vehicles on the road and the resulting longer queues (thus increasing the delays and time spent travelling under 10km/h), but it was not a large enough increase to cause a significant increase in additional vehicle stops. The steep increase from the 30-minute mark to the 45-minute mark occurred as a result of the vehicle spillback that arose as vehicle queues began backing up into the road, obstructing vehicles that were attempting to drive along the road. The gradual decrease over the remaining time period occurred once the spillback queues from the parking lot had cleared and vehicles that had not been obstructed began exiting the



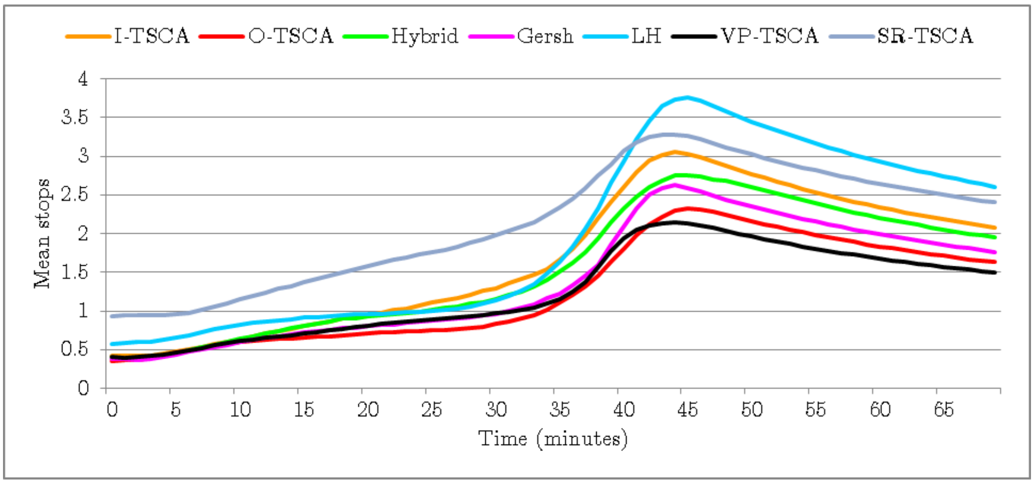
(a)

FIGURE 8.4: *The mean delay of vehicles in the scenario with a variable vehicle arrival rate.*



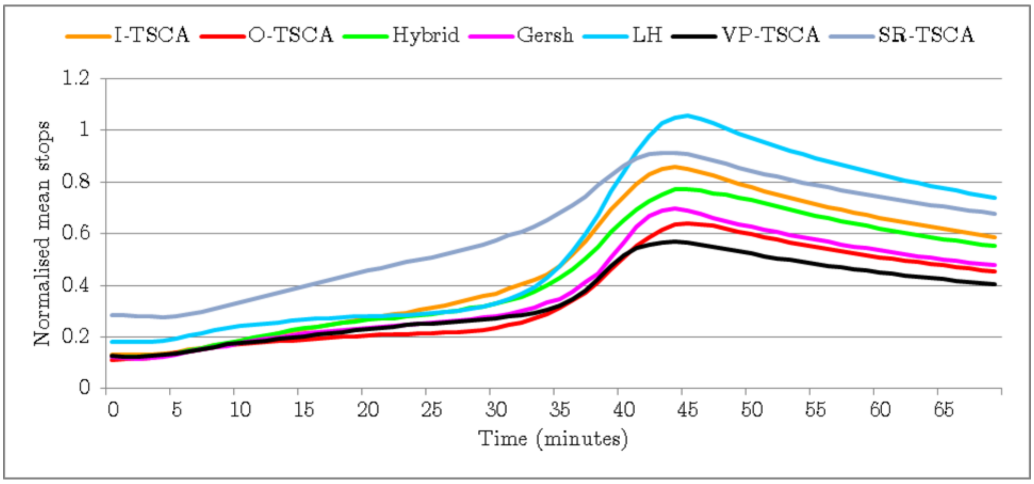
(a)

FIGURE 8.5: *The normalised mean delay of vehicles in the scenario with a variable vehicle arrival rate.*



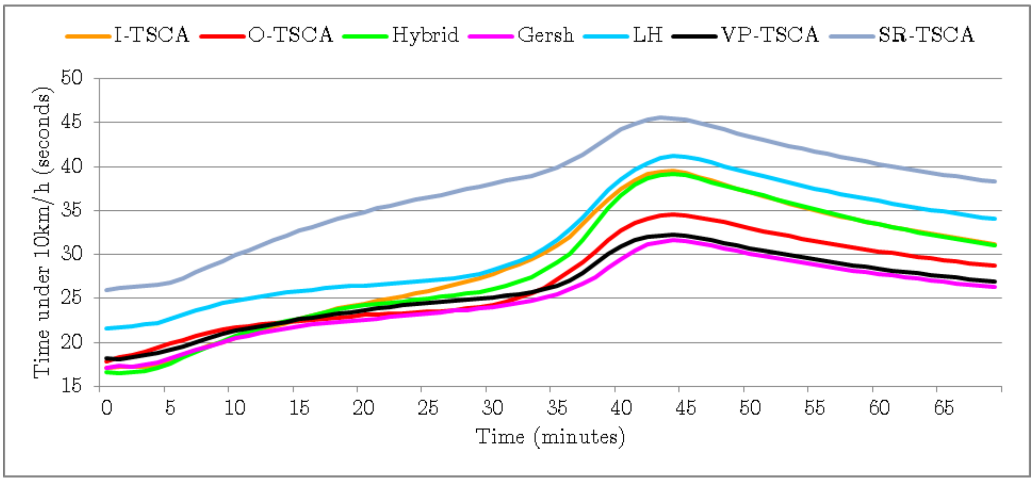
(a)

FIGURE 8.6: *The mean number of stops experienced by vehicles in the scenario with a variable vehicle arrival rate.*



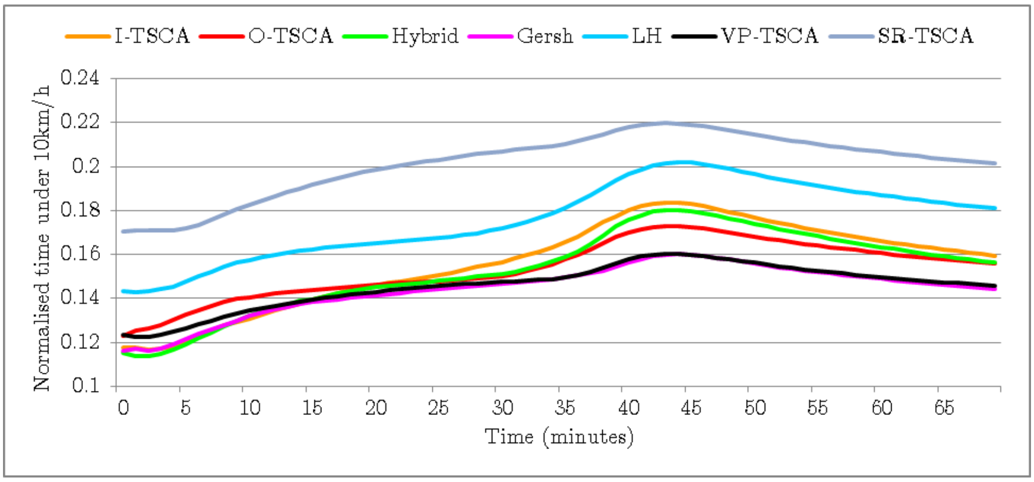
(a)

FIGURE 8.7: *The normalised mean number of stops experienced by vehicles in the scenario with a variable vehicle arrival rate.*



(a)

FIGURE 8.8: *The mean time vehicles spend travelling under 10km/h in the scenario with a variable vehicle arrival rate.*



(a)

FIGURE 8.9: *The normalised mean time vehicles spend travelling under 10km/h in the scenario with a variable vehicle arrival rate.*

network. By this time all vehicles travelling to the event had left the road network and thus the arrival rate had reduced back to $\lambda = 12$ vehicles per minute.

The ANOVA column in Table 8.8 indicates that for all six PMIs there are statistical differences between the means returned by the algorithms at a 5% level of significance. Furthermore, the outcome of the Levene test revealed that the algorithmic variances of the mean samples were statistically indistinguishable for PMI 1, PMI 5 and PMI 6, and therefore the Fisher LSD *post hoc* test was used in order to determine between which pairs of algorithmic outputs differences are statistically discernible in respect of these three PMIs. The Games-Howell *post hoc* test was used for this purpose in respect of the remaining three PMIs, namely PMI 2, PMI 3 and PMI 4, as the test indicated that there was a significant difference between the variances of the sample means for these PMIs at a 5% level of significance.

	Mean value								p-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA	ANOVA	Levene's Test
1	65.32	60.03	64.81	56.78	66.84	152.55	57.43	67.55	$<1 \times 10^{-17}$	1.82×10^{-1}
2	1.678	1.624	1.673	1.589	1.697	2.624	1.593	1.703	$<1 \times 10^{-17}$	3.84×10^{-2}
3	3.018	2.218	2.680	2.588	3.645	16.889	2.035	2.941	$<1 \times 10^{-17}$	4.09×10^{-4}
4	0.852	0.616	0.750	0.689	1.027	5.112	0.538	0.827	$<1 \times 10^{-17}$	4.65×10^{-6}
5	39.42	34.41	39.02	31.48	41.00	120.71	31.588	43.50	$<1 \times 10^{-17}$	2.18×10^{-1}
6	0.1838	0.1727	0.1802	0.1597	0.2014	0.4047	0.1594	0.2172	$<1 \times 10^{-17}$	1.27×10^{-1}

TABLE 8.8: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests in the context of a* 3×4 *grid of intersections with a variable arrival rate. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

Gersh, the VP-TSCA and the O-TSCA(n) statistically outperformed the other algorithms in terms of mean delay, obtaining values of 56.78 seconds, 57.43 seconds and 60.03 seconds, respectively, over the 70-minute period. Hybrid(n), the I-TSCA(n), LH and the SR-TSCA did not perform significantly differently from one another in this respect, but all outperformed Fixed at a 5% level of significance. Similar results were obtained for the normalised mean delay time, as Gersh and the VP-TSCA statistically outperformed all other algorithms in terms of the mean and the normalised mean delay time with the exception of the O-TSCA(n), from which they did not differ significantly. They did, however, outperform a larger number of algorithms than did the O-TSCA(n) in this respect. Gersh achieved a normalised mean delay PMI value of 1.589, indicating that vehicles under the control of Gersh spent a approximately of 59% longer in the road network as a result of delays, while LH (the worst performing algorithm, after Fixed) achieved a value of 1.697, suggesting that vehicles spent an additional of 70% longer in the network as a result of delays.

In respect of the mean and normalised mean number of stops experienced by vehicles travelling through the road network, the VP-TSCA returned the best results, obtaining corresponding PMI values of 2.035 and 0.538, respectively. The O-TSCA(n) was the next best performing algorithm in this respect, outperforming fewer algorithms than the VP-TSCA, but not differing significantly from the actual mean obtained by the VP-TSCA. Hybrid(n) and Gersh obtained the next best results in respect of both PMIs.

Gersh, the VP-TSCA and the O-TSCA(n) statistically outperformed the remaining five algorithms in respect of mean time spent travelling under 10km/h, achieving values of 31.48 seconds, 31.59 seconds and 34.41 seconds, respectively. The I-TSCA(n) and Hybrid(n) achieved the next best result in respect of this PMI. In respect of the normalised mean delay time spent travelling under 10km/h, the VP-TSCA and Gersh statistically outperformed all other algorithms, while

p-values of the Fisher LSD test: Mean delay time								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.77×10^{-2}	8.17×10^{-1}	1.50×10^{-4}	4.93×10^{-1}	$<1 \times 10^{-17}$	4.45×10^{-4}	3.15×10^{-1}
O-TSCA(n)		—	3.20×10^{-2}	1.44×10^{-1}	2.35×10^{-3}	$<1 \times 10^{-17}$	2.42×10^{-1}	8.03×10^{-4}
Hybrid(n)			—	3.59×10^{-4}	3.59×10^{-1}	$<1 \times 10^{-17}$	1.01×10^{-3}	2.17×10^{-1}
Gersh				—	9.01×10^{-6}	$<1 \times 10^{-17}$	7.71×10^{-1}	2.15×10^{-6}
LH					—	$<1 \times 10^{-17}$	3.10×10^{-5}	7.49×10^{-1}
Fixed						—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
VP-TSCA							—	7.94×10^{-6}
SR-TSCA								—
Mean	65.32	60.03	64.81	56.78	66.84	152.55	57.43	67.55

TABLE 8.9: *Differences in respect of the mean delay obtained for Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 with a variable arrival rate within the context of a 3 × 4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean delay time								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	3.47×10^{-1}	9.99×10^{-1}	5.03×10^{-3}	9.93×10^{-1}	1.34×10^{-11}	3.69×10^{-3}	9.55×10^{-1}
O-TSCA(n)		—	4.59×10^{-1}	6.99×10^{-1}	4.20×10^{-2}	6.07×10^{-12}	7.18×10^{-1}	9.24×10^{-3}
Hybrid(n)			—	8.71×10^{-3}	9.73×10^{-1}	1.29×10^{-11}	6.52×10^{-3}	8.87×10^{-1}
Gersh				—	8.78×10^{-5}	$<1 \times 10^{-17}$	9.99×10^{-1}	3.69×10^{-6}
LH					—	7.30×10^{-12}	4.14×10^{-5}	9.99×10^{-1}
Fixed						—	1.10×10^{-12}	$<1 \times 10^{-17}$
VP-TSCA							—	7.14×10^{-7}
SR-TSCA								—
Mean	1.678	1.624	1.673	1.589	1.697	2.624	1.593	1.703

TABLE 8.10: *Differences in respect of the normalised mean delay obtained for Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 with a variable arrival rate within the context of a 3 × 4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	4.01×10^{-1}	9.83×10^{-1}	9.71×10^{-1}	8.56×10^{-1}	$<1 \times 10^{-17}$	7.92×10^{-2}	9.99×10^{-1}
O-TSCA(n)		—	8.72×10^{-1}	9.82×10^{-1}	3.11×10^{-2}	1.08×10^{-12}	9.99×10^{-1}	3.50×10^{-1}
Hybrid(n)			—	9.99×10^{-1}	3.24×10^{-1}	1.11×10^{-12}	3.63×10^{-1}	9.91×10^{-1}
Gersh				—	3.48×10^{-1}	$<1 \times 10^{-17}$	7.84×10^{-1}	9.82×10^{-1}
LH					—	$<1 \times 10^{-17}$	3.85×10^{-3}	6.77×10^{-1}
Fixed						—	$<1 \times 10^{-17}$	4.68×10^{-13}
VP-TSCA							—	3.21×10^{-2}
SR-TSCA								—
Mean	3.018	2.218	2.680	2.588	3.645	16.89	2.035	2.941

TABLE 8.11: *Differences in respect of the mean number of stops obtained for Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 with a variable arrival rate within the context of a 3 × 4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	3.56×10^{-1}	9.78×10^{-1}	8.48×10^{-1}	8.64×10^{-1}	6.66×10^{-13}	2.75×10^{-2}	9.99×10^{-1}
O-TSCA(n)		—	8.57×10^{-1}	9.97×10^{-1}	2.42×10^{-2}	6.22×10^{-13}	9.78×10^{-1}	3.17×10^{-1}
Hybrid(n)			—	9.99×10^{-1}	3.04×10^{-1}	5.32×10^{-13}	1.67×10^{-1}	9.90×10^{-1}
Gersh				—	1.63×10^{-1}	4.63×10^{-13}	7.31×10^{-1}	8.77×10^{-1}
LH					—	$<1 \times 10^{-17}$	1.13×10^{-3}	6.67×10^{-1}
Fixed						—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
VP-TSCA							—	7.39×10^{-3}
SR-TSCA								—
Mean	0.852	0.616	0.750	0.689	1.027	5.112	0.538	0.827

TABLE 8.12: *Differences in respect of the normalised mean number of stops obtained for Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 with a variable arrival rate within the context of a 3 × 4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	p-values of the Fisher LSD test: Mean time spent under 10km/h							
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.82×10^{-2}	8.48×10^{-1}	2.07×10^{-4}	4.55×10^{-1}	$<1 \times 10^{-17}$	2.53×10^{-4}	5.44×10^{-2}
O-TSCA(n)		—	2.98×10^{-2}	1.65×10^{-1}	2.00×10^{-3}	$<1 \times 10^{-17}$	1.82×10^{-1}	2.40×10^{-5}
Hybrid(n)			—	4.20×10^{-4}	3.48×10^{-1}	$<1 \times 10^{-17}$	5.10×10^{-4}	3.46×10^{-2}
Gersh				—	9.88×10^{-6}	$<1 \times 10^{-17}$	9.57×10^{-1}	3.54×10^{-8}
LH					—	$<1 \times 10^{-17}$	1.25×10^{-5}	2.37×10^{-1}
Fixed						—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
VP-TSCA							—	4.67×10^{-8}
SR-TSCA								—
Mean	39.42	34.41	39.02	31.48	41.00	120.71	31.59	43.50

TABLE 8.13: *Differences in respect of the mean time spent travelling under 10km/h obtained for Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 with a variable arrival rate within the context of a 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	p-values of the Fisher LSD test: Normalised mean time spent under 10km/h							
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.00×10^{-2}	4.00×10^{-1}	4.56×10^{-8}	4.580×10^{-5}	$<1 \times 10^{-17}$	2.97×10^{-8}	1.50×10^{-13}
O-TSCA(n)		—	8.08×10^{-2}	2.48×10^{-3}	1.17×10^{-10}	$<1 \times 10^{-17}$	1.89×10^{-3}	$<1 \times 10^{-17}$
Hybrid(n)			—	2.68×10^{-6}	1.15×10^{-6}	$<1 \times 10^{-17}$	1.83×10^{-6}	6.66×10^{-16}
Gersh				—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	9.34×10^{-1}	$<1 \times 10^{-17}$
LH					—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	2.70×10^{-4}
Fixed						—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	0.1838	0.1727	0.1802	0.1597	0.2014	0.4047	0.1594	0.2172

TABLE 8.14: *Differences in respect of the normalised mean time spent travelling under 10km/h obtained for Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 with a variable arrival rate within the context of a 3×4 grid of intersections. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

not statistically differing from one another at a 5% level of significance. The O-TSCA(n) was found to outperform the I-TSCA(n), LH, the SR-TSCA and Fixed significantly, while once again Fixed was outperformed by all other algorithms.

Overall it appears that Gersh and the VP-TSCA performed the best with respect to the PMIs overall, as they were not outperformed by another algorithm. The O-TSCA(n) also performed comparatively well, as it was only ever statistically outperformed in one of the PMI values. LH and Fixed did not perform well overall, as neither algorithm significantly improved upon a single algorithm in respect of any of the PMIs. The very poor performance of Fixed is attributed to its very rigid nature. Since the green splits and cycle times were calculated to account for the initial traffic flow of the road network ($\lambda = 12$), the algorithm struggled to serve the vehicles effectively once the arrival rate increased, particularly as the number of right turning vehicles increased, which resulted in vehicles backing up into downstream intersections. There was a considerable amount of variance in the values returned for this scenario across all PMIs as a result of the fluctuating arrival rates and hence, it was more difficult to obtain significant differences between PMI means at a 5% level of significance than in previous scenarios.

8.2 A corridor road network

Two different scenarios involving a corridor road network are investigated in this section. The first scenario involves the scaling of a corridor network by comparing the results returned by the algorithms for corridors of different sizes. The second scenario involves varying the distances between consecutive intersections in the corridor, in order to determine whether the spacing of intersections plays a role in the relative performance of the algorithms.

8.2.1 Scaling of a road corridor

Corridors containing various numbers of intersections are considered in this section in order to determine whether the scaling of corridor sizes has an effect on the performance of the fixed control strategy and the seven self-organising algorithms. This is done in order to ensure that as the network becomes larger and more complex, the algorithms still perform proportionally well. Simulation results were obtained and a statistical analysis was carried out for an arrival rate of $\lambda = 20$ vehicles per minute, while all intersections were identical and equally spaced in each of three corridors.

The three corridors involved in this experiment include a four-, six- and eight-intersection corridor. The scaling of the algorithms was measured according to the increase in PMI values from the four-intersection corridor to the six-intersection corridor and comparing it to the increase found from the six-intersection corridor to the eight-intersection corridor. The relative performance of the algorithms over the three different corridor topologies is also reported.

The resulting PMI values were influenced by both the number of intersections as well as the algorithms that were employed at a given time, and therefore the results are reported slightly differently than in previous sections in order to take into account the effects of both of these independent variables. The *post hoc* tests performed take into account both the algorithms and the number of intersections during the calculation of significant differences. The results are given in the form of graphs and tables that include the first and second differences[1].

[1]The first or initial difference refers to the difference between the results of the four and six-intersection corridor, while the second difference refers to the difference between the results of the six and eight-intersection corridor.

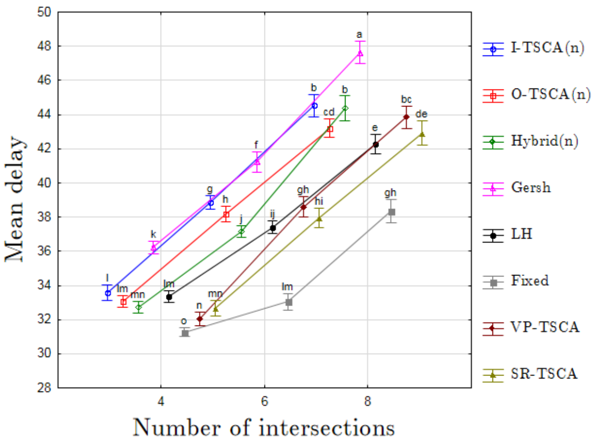
Simulation results

In respect of mean delay, it appears that the PMI values of all algorithms, with the exception of Hybrid(n) and Fixed, increased relatively linearly with an increase in the number of intersections, as shown in Figure 8.10(a) and Table 8.15. The growth of the mean delay experienced by vehicles appears to accelerate with the addition of intersections for both Hybrid(n) and Fixed. The relative performances of the algorithms are very similar in the four-intersection corridor, with the exception of Gersh which performed significantly worse, and Fixed, which performed significantly better. Differences in respect of mean delay returned by the algorithms become more pronounced in the six-intersection corridor, where Fixed significantly outperformed all algorithms. Hybrid(n) went from one of the best performing algorithms in the six-intersection corridor to the second-worst performing algorithm in the eight-intersection corridor, only outperforming Gersh and not differing significantly from the I-TSCA(n) or the VP-TSCA. This is not surprising as it is a direct result of the accelerated growth in first and second differences shown in Table 8.15. Fixed obtained the best results over all the scenarios, despite the increase in PMI difference from the six- to eight-intersection corridor.

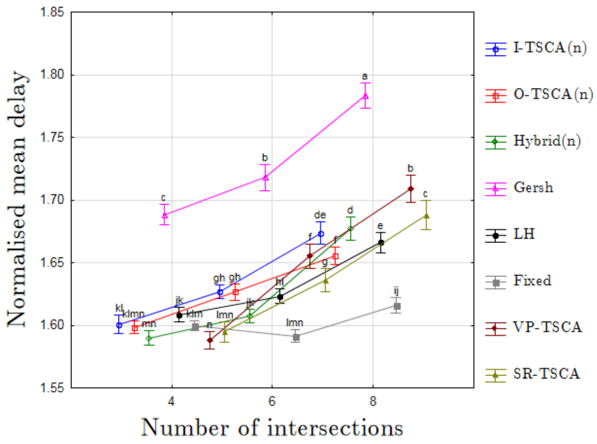
The O-TSCA(n) once again exhibited a linear relationship between the normalised mean delay and the number of intersections, as can be seen in Table 8.16, while the I-TSCA(n), LH and Gersh all experienced a similar, larger margin of increase during the six-to-eight-intersection difference. Hybrid(n) obtained a substantially larger margin of increase for the six-to-eight-intersection difference, obtaining an increase more than four times larger than the initial difference, as may be seen in Figure 8.10(b). Hybrid(n) was one of the best-performing algorithms in the four-intersection corridor as it was not outperformed by another algorithm. In the context of the six-intersection corridor, Hybrid(n) significantly outperformed every other algorithm with the exception of Fixed, while its performance dropped significantly in the eight-intersection corridor where it was outperformed by the O-TSCA(n), LH and Fixed. Interestingly, the VP-TSCA and the SR-TSCA obtained relatively good results in the four-intersection corridor as they were not outperformed by another algorithm, while they were the second- and third-worst performing algorithm, respectively, in terms of the eight-intersection corridor. Fixed showed no difference in results between four and six-intersection corridors, while there was a slight increase in the PMI for the eight-intersection corridor.

The O-TSCA(n) maintained a linear relationship between the mean number of stops made by vehicles and the number of intersections, although it did exhibit the second largest initial difference percentage increase over all the algorithms, as shown in Table 8.17. LH obtained small first and second differences, although it was less linear than the O-TSCA(n), obtaining a larger second difference than first difference. Hybrid(n) once again achieved the steepest increase in differences from a first difference of 20.76% to a second difference of 53.16%, indicating that the number of stops experienced by vehicles under the control of this algorithm increased dramatically with an increase in road network size. The O-TSCA(n) significantly outperformed all the algorithms with the exception of Hybrid(n) and the VP-TSCA (from which it did not differ significantly) in respect of the mean number of stops made by vehicles in the four-intersection corridor. In the six-intersection corridor, Hybrid(n) was not significantly outperformed by any algorithm, but outperformed the I-TSCA(n), Gersh, the VP-TSCA and the SR-TSCA. In the eight-intersection corridor LH and the O-TSCA(n) were the best-performing algorithms, significantly outperforming all the other algorithms at a 5% level of significance.

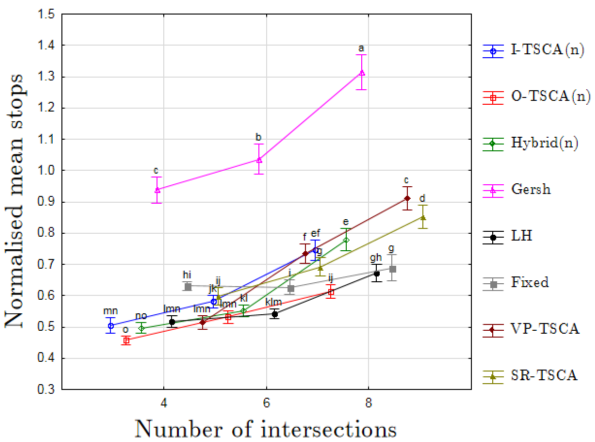
Hybrid(n), LH and Fixed achieved the largest increase in differences for normalised mean number of vehicle stops, with Hybrid(n) obtaining a second difference almost four times larger than the first difference, as shown in Figure 8.18. The O-TSCA(n) obtained constant increases and thus



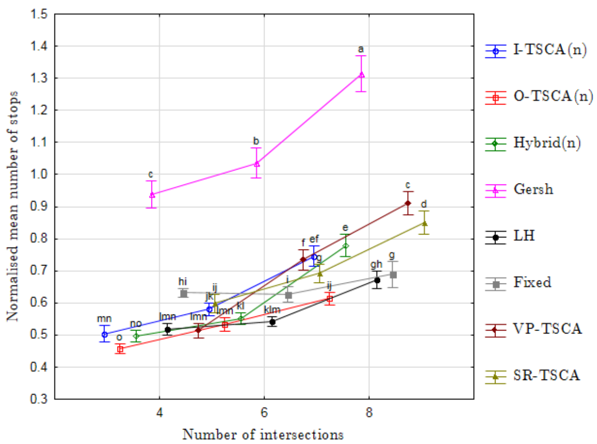
(a) *Mean delay time*



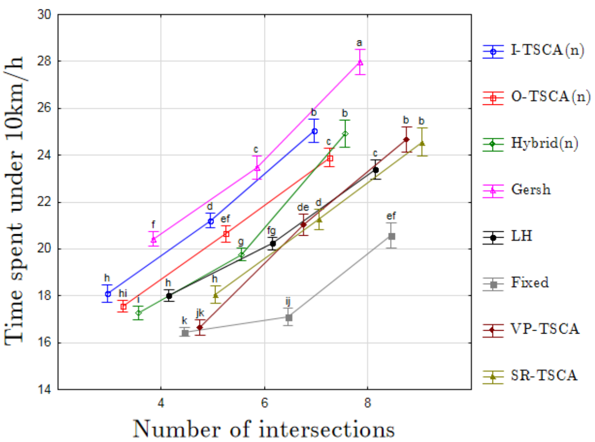
(b) *Normalised mean delay time*



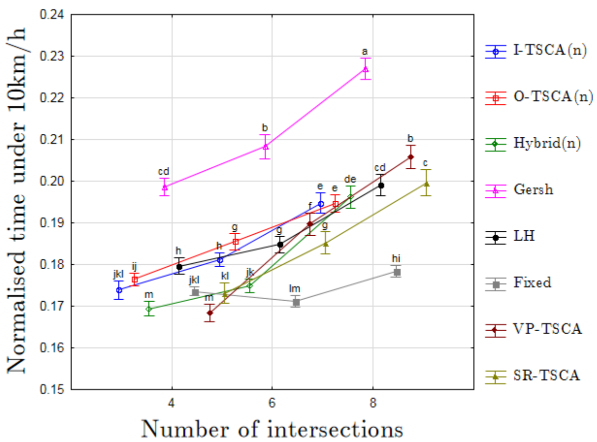
(c) *Mean number of stops*



(d) *Normalised mean number of stops*



(e) *Mean time spent under 10km/h*



(f) *Normalised mean time spent under 10km/h*

FIGURE 8.10: *PMI results for the self-organising algorithms in a four-intersection, six-intersection and eight-intersection road corridor.*

Algorithm	Mean delay time						
	Intersection			First difference		Second difference	
	4	6	8	Value	%	Value	%
I-TSCA(n)	33.58	38.88	44.52	5.30	15.78	5.64	14.51
O-TSCA(n)	33.06	38.17	43.20	5.11	15.46	5.03	13.18
Hybrid(n)	32.72	37.16	44.35	4.44	13.57	7.19	19.35
Gersh	36.22	41.22	47.64	5.00	13.80	6.42	15.57
LH	33.37	37.41	42.27	4.04	12.11	4.86	12.99
Fixed	31.27	33.05	38.37	1.78	5.690	5.32	16.10
VP-TSCA	32.05	38.60	43.85	6.55	20.44	5.25	13.60
SR-TSCA	32.67	37.96	42.92	5.29	16.19	4.96	13.07

Table 8.15: *The mean delay differences obtained between a four-intersection and a six-intersection corridor, compared to the differences between a six-intersection and an eight-intersection corridor.*

Algorithm	Normalised mean delay time						
	Intersection			First difference		Second difference	
	4	6	8	Value	%	Value	%
I-TSCA(n)	1.601	1.627	1.674	0.026	1.62	0.047	2.89
O-TSCA(n)	1.598	1.627	1.655	0.029	1.81	0.028	1.72
Hybrid(n)	1.590	1.607	1.677	0.017	1.07	0.070	4.36
Gersh	1.688	1.718	1.783	0.030	1.78	0.065	3.78
LH	1.609	1.623	1.666	0.014	0.87	0.043	2.65
Fixed	1.599	1.591	1.616	−0.080	0.50	0.025	1.57
VP-TSCA	1.588	1.655	1.709	0.067	4.22	0.054	3.26
SR-TSCA	1.595	1.636	1.688	0.041	2.57	0.052	3.18

Table 8.16: *The normalised mean delay differences obtained between a four-intersection and a six-intersection corridor, compared to the differences between a six-intersection and an eight-intersection corridor.*

Algorithm	Mean number of stops						
	Intersection			First difference		Second difference	
	4	6	8	Value	%	Value	%
I-TSCA(n)	0.864	1.097	1.475	0.233	26.97	0.378	34.46
O-TSCA(n)	0.768	0.991	1.264	0.223	29.04	0.273	27.55
Hybrid(n)	0.838	1.012	1.550	0.174	20.76	0.538	53.16
Gersh	1.389	1.678	2.234	0.289	20.81	0.556	33.13
LH	0.850	0.969	1.236	0.119	14.00	0.267	27.55
Fixed	0.957	1.044	1.393	0.087	9.09	0.349	33.43
VP-TSCA	0.837	1.271	1.646	0.434	51.85	0.375	29.50
SR-TSCA	1.007	1.248	1.563	0.241	23.93	0.315	25.24

Table 8.17: *The mean number of stops differences obtained between a four-intersection and a six-intersection corridor, compared to the differences between a six-intersection and an eight-intersection corridor.*

Algorithm	Normalised mean number of stops Intersection 4	6	8	First difference Value	%	Second difference Value	%
I-TSCA(n)	0.504	0.581	0.745	0.077	15.28	0.164	28.23
O-TSCA(n)	0.458	0.533	0.614	0.075	16.38	0.081	15.20
Hybrid(n)	0.497	0.551	0.779	0.054	10.87	0.228	41.38
Gersh	0.938	1.036	1.314	0.098	10.45	0.278	26.83
LH	0.517	0.542	0.672	0.025	4.840	0.130	23.99
Fixed	0.632	0.627	0.689	−0.005	0.790	0.062	9.890
VP-TSCA	0.514	0.734	0.911	0.220	42.80	0.177	24.11
SR-TSCA	0.598	0.692	0.851	0.094	15.72	0.159	22.98

TABLE 8.18: *The normalised mean number of stops differences obtained between a four-intersection and a six-intersection corridor, compared to the differences between a six-intersection and an eight-intersection corridor.*

Algorithm	Mean time spent under 10km/h Intersection 4	6	8	First difference Value	%	Second difference Value	%
I-TSCA(n)	18.09	21.21	25.04	3.12	17.25	3.83	18.06
O-TSCA(n)	17.56	20.64	23.91	3.08	17.54	3.27	15.84
Hybrid(n)	17.25	19.77	24.92	2.52	14.61	5.15	26.05
Gersh	20.42	23.47	27.98	3.05	14.94	4.51	19.22
LH	18.01	20.24	23.38	2.23	12.38	3.14	15.51
Fixed	16.46	17.11	20.56	0.65	3.950	3.45	20.16
VP-TSCA	16.65	21.04	24.66	4.39	26.37	3.62	17.21
SR-TSCA	18.06	21.26	24.56	3.20	17.72	3.30	15.52

TABLE 8.19: *The mean time spent under 10km/h differences obtained between a four-intersection and a six-intersection corridor, compared to the differences between a six-intersection and an eight-intersection corridor.*

Algorithm	Normalised mean time spent under 10km/h Intersection 4	6	8	First difference Value	%	Second difference Value	%
I-TSCA(n)	0.1739	0.1812	0.1948	0.0073	4.20	0.0136	7.51
O-TSCA(n)	0.1765	0.1855	0.1947	0.0090	5.10	0.0092	4.96
Hybrid(n)	0.1694	0.1750	0.1962	0.0056	3.31	0.0212	12.1
Gersh	0.1986	0.2083	0.2270	0.0097	4.88	0.0187	8.98
LH	0.1796	0.1848	0.1991	0.0052	2.90	0.0143	7.74
Fixed	0.1735	0.1711	0.1784	−0.002	1.40	0.0070	4.27
VP-TSCA	0.1684	0.1897	0.2059	0.0213	12.7	0.0162	8.54
SR-TSCA	0.1731	0.1852	0.1996	0.0121	6.99	0.0144	7.78

TABLE 8.20: *The normalised mean time spent under 10km/h differences obtained between a four-intersection and a six-intersection corridor, compared to the differences between a six-intersection and an eight-intersection corridor.*

a linear relationship emerged for the normalised mean number of stops experienced by vehicles, as illustrated in Figure 8.10(d). In a four-intersection corridor, the results for the normalised mean number of stops show that the relative performances of the I-TSCA(n), the O-TSCA(n), Hybrid(n), LH and the VP-TSCA are all very similar, although the O-TSCA(n) appears to be the superior algorithm as it outperformed more algorithms. Similar results may be observed in the context of the six-intersection corridor for the O-TSCA(n), Hybrid(n) and LH, which did not perform statistically differently from one another. For the eight-intersection corridor, however, the differences between the algorithmic performances were more significant, with the O-TSCA(n) significantly outperforming the other algorithms, followed by LH and Fixed. Once again, the performance of Hybrid(n) dropped dramatically from the six-intersection corridor to the eight-intersection corridor.

It is clear from Figure 8.10(e) and the relatively small change in first and second differences in Table 8.19 that the mean time spent travelling under 10km by vehicles under control of the O-TSCA(n), LH, the I-TSCA(n) and the SR-TSCA increased linearly with respect to an increase of road network size. Hybrid(n), Fixed and Gersh, once again returned substantially larger differences than the other algorithms. VP-TSCA and Fixed were statistically the best performing algorithms in the four-intersection corridor, outperforming all algorithms and not differing significantly from one another. In the six-intersection corridor, Fixed significantly outperformed all other algorithms at a 5% level of significance. The increase in PMI values for Fixed from six to eight intersections is very large, although it is still the best performing algorithm.

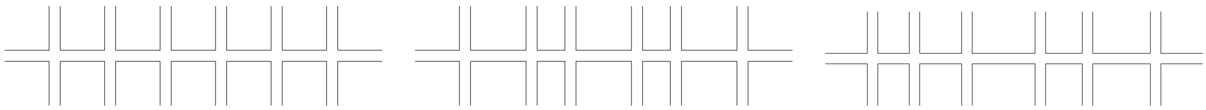
In respect of the normalised mean delay time vehicles spend travelling under 10km/h, the O-TSCA(n) and the SR-TSCA were the only algorithms to achieve a relatively linear PMI value increase. Hybrid(n), Fixed and LH returned second differences about 3 time larger than their initial difference, indicating that the performances of these algorithms deteriorated considerably with an increase in the number of intersections in the road network. The margin of increase in differences was similar for the I-TSCA(n) and Gersh, as may be seen in Table 8.20, although the relative performances of these two algorithms were very different, as shown in Figure 8.10(f). Hybrid(n) and the VP-TSCA significantly outperformed all other algorithms in terms of normalised mean time travelling under 10km/h in the four-intersection corridor, while Fixed was the best performing algorithm in the six-intersection corridor. The performance of Hybrid(n) significantly deteriorated in the eight-intersection corridor, only outperforming the SR-TSCA, the VP-TSCA and Gersh. In the eight-intersection corridor, the I-TSCA(n), the O-TSCA(n), Hybrid(n), LH and the SR-TSCA did not perform significantly differently from one another, while they all outperformed Gersh by a considerable margin, and were all outperformed by Fixed by a considerable margin.

The O-TSCA(n) and LH obtained the most favourable results overall, achieving the best balance between small PMI values and a small growth rate of PMI values relative to an increase in the number of intersections in the corridor. While Fixed returned the best PMI values overall, these values seem to grow exponentially with an increase in number of intersections. This is because the longer the corridor of intersections, the larger the platoon of vehicles forming the green waves become and once the platoons become very large, they are not all served during the assigned green time, resulting in the platoon being separated. Hybrid(n), on the other hand, while achieving very promising results for the four-intersection and six-intersection corridor, suffered a dramatic drop in performance when the number of intersections was increased, as is clear in Figures 8.10(a)–8.10(f). While Gersh did not obtain difference increases as large as those exhibited by Hybrid(n), the PMI values were already so much larger than those of the other algorithms, that it performs poorly throughout all three corridors.

8.2.2 Varying distances between consecutive intersections

The scenario considered in this section involves a six-intersection corridor in which the spacing between consecutive intersections is varied. The algorithmic performances were compared with one another as well as with their own performances in three different intersection spacing layouts. The arrival rates in this section were defined in such a way that sizeable platoons of vehicles enter the road network periodically in the west-to-east and east-to-west directions. This scenario was designed to test how the well the algorithms are able to achieve coordination between intersections, which is primarily reflected by the average number of stops made by vehicles.

The first of the three spacing variations of the six-intersection corridor, referred to as *equal* and shown in Figure 8.11(a), involved a corridor containing equally spaced intersections 360 metres apart. The second variation, referred to as *multiple*, consisted of two alternating different intersection distances, such that one of the distances is a multiple of the other, as shown in Figure 8.11(b). The third and final spacing layout consisted of intersections at irregular distances from one another, such that no two distances are the same or are multiples of each other, as shown in Figure 8.11(c).



(a) *Equal spacing* (b) *Multiple spacing* (c) *Uneven spacing*

FIGURE 8.11: *The three different spacing layouts considered for a six-intersection corridor.*

The horizontal arrival rate was set to $\lambda = 60$ vehicles per minute, for a 20-second interval, alternating with a 20-second interval in which the arrival rate was $\lambda = 0$ vehicles per minute. A constant arrival rate of $\lambda = 5$ was set for vehicles generated in the vertical direction. The generation of large platoons (consisting of approximately 20 vehicles) in the horizontal direction, together with a light traffic flow in the vertical direction, gave the algorithms the potential to allocate green times effectively.

Simulation results

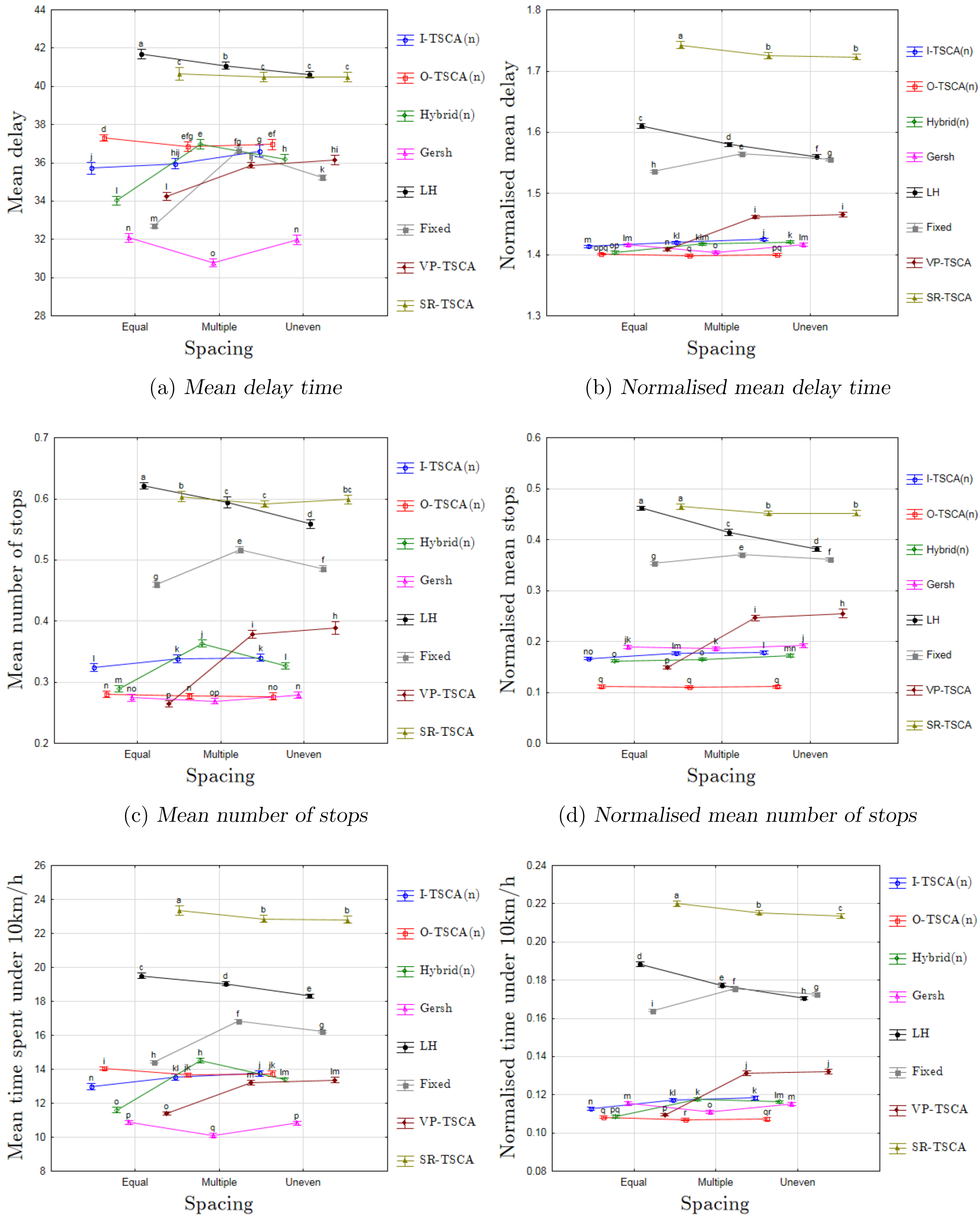
It becomes immediately clear from the graphs in Figure 8.12 that the PMIs do not exhibit a general increase with an increase in randomness regarding the spacing of intersections. There are, however, certain trends visible in some of the algorithmic results which may be indirect results of the various spacing layouts. The multiple layout, for example, contained much smaller distances between some of the intersections than in the other layouts, which may have played a role in the effectiveness of the algorithms.

In respect of the mean delay time, Gersh significantly outperformed all other algorithms, obtaining similar results in the equal and uneven scenario, but a significantly better result in the multiple scenario, improving upon the equal and uneven scenario by just over one second, as shown in Figure 8.12(a). This was due to the strong coordination achieved in both directions between the two pairs of closely spaced intersections, where vehicles travelling through the first of these intersections, were rarely required to slow at the following intersection. The I-TSCA(n) and the O-TSCA(n) achieved relatively linear PMI results in respect of the spacing layout — the differences are practically insignificant. Hybrid(n) performed significantly better than the I-TSCA(n), the O-TSCA(n), LH and the SR-TSCA under equal intersection spacing, but worsened significantly in the multiple spacing case, only outperforming LH and the SR-TSCA in this regard. This suggests that Hybrid(n) did not achieve coordination between the closer spaced intersections. This may also explain why the algorithm performance improved from the multiple to the uneven spacing, since the uneven spacing layout also contained closely spaced intersections, but not as closely spaced as those in the multiple layout. LH was found to follow a unique trend in respect of mean delay time, improving upon the PMI values the less consistent the intersection spacing. While the differences in PMI values achieved by LH over the three layouts were statistically significant, they are not practically significant, given that the differences were approximately 0.6 seconds. The performance of the VP-TSCA worsened as the intersection spacing became more random, suggesting that this algorithm is reliant on regular spacing of intersections in order to operate effectively. In contrast to the VP-TSCA, the SR-TSCA maintained similar results throughout all the spacing scenarios, although the individual results were not good in comparison with the other algorithms. Fixed obtained very good results for the even case, but this worsened as the spacing became more random, since green waves were unable to form in both directions when intersection spacings were not equal.

The O-TSCA(n) obtained very similar results in the equal, multiple and uneven spacing layouts in terms of normalised mean delay time. These results did not differ significantly from one another, suggesting that the performance of the O-TSCA(n) is not dependent on the spacing between consecutive intersections. The O-TSCA(n) was also the overall best-performing algorithm in respect of normalised mean delay time, as it was not outperformed by any other algorithm over all three spacing layouts in this respect. The I-TSCA(n), Hybrid(n) and Gersh PMI result differences over the various spacing layouts were so small that they are considered practically insignificant, as may be seen in Figure 8.12(b). The VP-TSCA performed well in the equal scenario but did worsen as the intersection spacing became less uniform, although it did outperform Fixed, LH and the SR-TSCA. LH and the SR-TSCA performed significantly worse than the other six algorithms in all intersection spacing layouts, although it appeared that the performance of LH significantly increased with an increase of intersection non-uniformity.

The O-TSCA(n) once again achieved consistent results across all three spacing layouts in respect of the mean number of stops; none of the three means were significantly different from one another. Gersh obtained a similar consistency across the three spacing layouts — the mean obtained for equal spacing did not differ significantly from the multiple or uneven layout means. The I-TSCA(n) achieved a smaller mean number of stops value in the equal spacing layout than in the multiple and uneven spacing layouts, although these two values did not differ significantly from one another. Hybrid(n) behaved similarly to how it did in respect of mean delay time, with the multiple spacing obtaining the largest mean as a result of the lack of coordination achieved between the closer spaced intersections, causing vehicles to stop more often than in the equal or uneven spacing layout. The O-TSCA(n) and Gersh statistically outperformed all other algorithms in respect of the mean number of stops over all three spacing layouts, as well as in terms of mean consistency, as shown in Figure 8.12(c). The values returned by the SR-TSCA did not differ significantly over the three scenarios, but it was the worst-performing algorithm under uneven intersection spacing conditions. Once again the performance of Fixed worsened in the multiple and uneven scenarios as a result of green waves only being able to form in one direction.

In respect of the normalised number of stops, the O-TSCA(n) and Gersh obtained differences between spacing layouts that are statistically insignificant as is evident from the straight line trends in Figure 8.12(d). The I-TSCA(n) and Hybrid(n) obtained some differences which are statistically significant, although the differences are not practically significant. LH and the SR-TSCA once again exhibited similar trends as before in terms of the previous PMIs, while remaining statistically the two worst-performing algorithms over all three road layouts. The



(a) *Mean delay time*

(b) *Normalised mean delay time*

(c) *Mean number of stops*

(d) *Normalised mean number of stops*

(e) *Mean time spent under 10km/h*

(f) *Normalised mean time spent under 10km/h*

FIGURE 8.12: *PMI results for the self-organising algorithms in a six-intersection corridor where distances between intersections are equal, multiples of some value, or uneven.*

Spacing	Algorithm							
	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
Equal	35.72	37.30	34.02	32.08	41.68	32.70	34.25	40.65
Multiple	35.96	36.86	36.98	30.77	41.09	36.66	35.88	40.48
Uneven	36.61	36.97	36.21	31.97	40.62	35.25	36.14	40.48

Table 8.21: *The mean delay in a six-intersection corridor where distances between intersections are equal, multiples of some value, or uneven.*

Spacing	Algorithm							
	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
Equal	1.413	1.401	1.403	1.416	1.611	1.536	1.409	1.742
Multiple	1.420	1.399	1.417	1.404	1.580	1.565	1.462	1.726
Uneven	1.425	1.400	1.421	1.416	1.560	1.556	1.466	1.723

Table 8.22: *The normalised mean delay in a six-intersection corridor where distances between intersections are equal, multiples of some value, or uneven.*

Spacing	Algorithm							
	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
Equal	0.324	0.280	0.289	0.275	0.621	0.459	0.264	0.604
Multiple	0.339	0.277	0.364	0.269	0.594	0.516	0.378	0.591
Uneven	0.340	0.277	0.327	0.279	0.559	0.486	0.389	0.599

Table 8.23: *The mean number of stops in a six-intersection corridor where distances between intersections are equal, multiples of some value, or uneven.*

Spacing	Algorithm							
	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
Equal	0.167	0.111	0.162	0.189	0.462	0.354	0.149	0.465
Multiple	0.177	0.111	0.165	0.186	0.414	0.371	0.246	0.452
Uneven	0.179	0.111	0.172	0.192	0.382	0.361	0.255	0.452

Table 8.24: *The normalised mean number of stops in a six-intersection corridor where distances between intersections are equal, multiples of some value, or uneven.*

Spacing	Algorithm							
	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
Equal	12.97	14.05	11.60	10.87	19.50	14.43	11.42	23.35
Multiple	13.55	13.67	14.53	10.11	19.05	16.82	13.21	22.86
Uneven	13.77	13.74	13.39	10.82	18.31	16.22	13.37	22.81

Table 8.25: *The mean time spent under 10km/h in a six-intersection corridor where distances between intersections are equal, multiples of some value, or uneven.*

Spacing	Algorithm							
	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
Equal	0.1129	0.1083	0.1085	0.1154	0.1884	0.1641	0.1096	0.2200
Multiple	0.1171	0.1069	0.1176	0.1109	0.1774	0.1757	0.1314	0.2154
Uneven	0.1183	0.1074	0.1162	0.1154	0.1706	0.1721	0.1322	0.2135

Table 8.26: *The normalised mean time under 10km/h in a six-intersection corridor where distances between intersections are equal, multiples of some value, or uneven.*

comparative performances of the algorithms indicated that the O-TSCA(n) was significantly the best performing algorithm over all spacing layouts. Hybrid(n) was statistically the second best-performing algorithm, outperforming Gersh and LH, but not differing significantly from the I-TSCA(n). While VP-TSCA obtained a relatively good result in terms of equal spacing of intersections, its performance once again worsened over the other two scenarios.

Relatively similar results were obtained for the mean time spent travelling under 10km/h as for mean delay, with Gersh achieving significantly better results for the multiple scenario due to the coordination it achieved between the closely spaced intersections, while significantly outperforming the other algorithms over all three spacing layouts. The I-TSCA(n) and the O-TSCA(n) obtained no significant difference in terms of this PMI in the multiple and uneven spacing layouts. Hybrid(n), the VP-TSCA and Fixed all performed significantly worse in the multiple spacing layout than in the equal or uneven spacing, while LH and the SR-TSCA once again followed the same trend as before, performing significantly worse than the other algorithms. While Hybrid(n) significantly outperformed the I-TSCA(n) and the O-TSCA(n) in the equal spacing layout, it was significantly outperformed by these two algorithms in the multiple and uneven spacing layouts. The VP-TSCA performed better in respect of this PMI value under multiple and uneven spacing layouts as it was only outperformed by Gersh.

The O-TSCA(n) and Hybrid(n) obtained the most favourable results in terms of the normalised mean time spent travelling under 10km/h in the equal spacing layout, while in the multiple spacing layout the O-TSCA(n) remained the best performing algorithm and Hybrid(n) performed significantly worse than in the equal spacing layout. Hybrid(n) was outperformed by the O-TSCA(n), the VP-TSCA and Gersh, while not significantly outperforming the I-TSCA(n) in the multiple spacing layout. Hybrid(n) improved slightly in the uneven spacing layout, statistically outperforming the I-TSCA(n) and the VP-TSCA, but not differing significantly from Gersh. Gersh was found to perform better in the multiple spacing layout in respect of this PMI, while the opposite occurred with Hybrid(n).

The O-TSCA(n) and Gersh achieved the overall best PMI results over the three spacing layouts. The O-TSCA(n) achieved the most consistent results over all the three of the layouts, indicating that the performance of the algorithm is not influenced by the spacing of intersections. Gersh also achieved consistent results for the equal and uneven intersection spacing layouts, for during the multiple spacing layout an improved result was obtained, indicating that Gersh may prove to be more effective if implemented in closely spaced intersections. It was also observed during the simulation runs that Gersh obtained excellent coordination between consecutive intersections, allowing platoons to travel through the intersections without slowing or separation of the platoons. Hybrid(n) performed well for equal spaced intersections, yet performed relatively poorly when the distances between consecutive intersections decreased. Hybrid(n) tended to separate the tails of platoons (as a result of the I-TSCA(n) component) and the minimum green time was not effective for the vertical direction, since few vehicles were arriving from that direction. The I-TSCA(n) performed relatively consistently over the three scenarios in comparison with the other algorithms, although it was generally outperformed by other algorithms as the I-TSCA(n) had a tendency to separate the tail of a platoon during service. There was no clear preferential treatment for the horizontal direction, which was necessary for effective green time allocation in this scenario, since a larger number of vehicles were approaching from the horizontal direction than from the vertical direction. LH did not perform well in comparison with other algorithms in this scenario due to the extremely short green times LH is known to allocate to light traffic flows. Vehicles in the vertical directions were often forced to wait long periods of time before making it through an intersection, since the LH favoured approaches with heavier traffic flows, as was seen in the previous sections where heavier traffic conditions were considered in both the horizontal and vertical directions. The SR-TSCA also had a tendency to neglect the light traffic flow emanating from approaching side streets, giving preferential treatment to the main flow through the corridor. Fixed performed better in the context of an equal spacing of intersections, since green waves were able to travel in two directions when intersection spacing was equal. Once the spacing was no longer uniform, green waves were only able to travel through the intersections unimpeded in one direction.

8.3 Chapter summary

Two different scenarios involving a 3×4 grid of intersections were considered in §8.1. The first of these involved a missing road link in §8.1.1, in order to measure how well the algorithms respond to a road closure. The results for the closed road scenario revealed that Gersh, Hybrid(n) and the VP-TSCA were the best performing algorithms overall. The second scenario involved a varying vehicle arrival rate that represented abnormal traffic induced by a large event such as a concert in §8.1.2. In this scenario, Gersh and the VP-TSCA were once again the superior algorithms, followed by O-TSCA(n).

The final two scenarios were considered in §8.2, in the context of a corridor road network. The first of these two scenarios involved the scaling of a corridor by comparing the algorithmic performances in four-intersection, six-intersection and eight-intersection corridors. The results revealed that the O-TSCA(n), LH and Fixed achieved the smallest PMI values overall, while obtaining relatively constant PMI value growth with an increase in number of intersections. The final scenario involved a six-intersection corridor in which distances between consecutive intersections were varied. The O-TSCA(n) and Gersh were found to achieve the best overall PMI values in this case, as well as the most favourable behaviour over the various intersection spacing layouts, indicating that their performances are not heavily reliant on the spacing between consecutive intersections in a corridor.

CHAPTER 9

Algorithmic performance in a real-world scenario

Contents

This chapter is devoted to a comparison of the eight traffic signal control algorithms of §5.1, §5.3 and §7 in a more realistic road network model than the corridor and grid network contexts of §6. Various elements of the road network are described in §9.1 and the results of a comparison of the relative algorithmic performances within the context of this road network are presented for two different times during the day in §9.2. The chapter closes in §9.3 with a summary of its contents.

9.1 Road network details

The real road network considered in this section consists of a provincial route, referred to as the R44, in the South African Western Cape that connects the towns of Piketberg and Kleinmond via Wellington, Stellenbosch and Somerset West. The particular section of this road under consideration is a corridor of eight consecutive signal-controlled intersections and three intermediate stop-controlled intersections within the town of Stellenbosch, as shown in Figure 9.1.

In a previous experiment carried out by the Department of Civil Engineering at Stellenbosch University [81], traffic counts were conducted at the eight signal-controlled intersections shown in Figure 9.1 for all traffic movements through the intersections. These traffic data were collected on a Tuesday from 6am until 6pm in April 2013. Such data for the three side streets that are not controlled by traffic lights were not, however, included in the experiment. These missing data were therefore obtained by the author as a result of counting vehicles and their respective movements through the three intersections on a Tuesday in June 2017 and interpolating the data for different periods of the day. In particular, traffic data were collected during 15-minute

FIGURE 9.1: *A screen shot of the R44 road network as implemented in the Anylogic simulation software suite [3]. The numbered red circles indicate intersections at which all approaches are controlled by traffic signals, while the blue circles indicate intersections at which side street approaches are controlled by a stop sign and vehicles travelling along the R44 are not required to stop. The black circles with crosses finally represent road closures.*

intervals at the intersections between the R44 and Alexander Road, Molteno Road and Bell Road between 11am and 12pm. These vehicle counts may be found in Tables 9.1–9.3.

Approach	Movement	Recorded	Hourly
R44	Right	108	432
	Left	20	80
Alexander	Right	14	56
	Left	87	348

TABLE 9.1: *Vehicle turn counts from the R44 into Alexander Road, and from Alexander Road into the R44, during a 15-minute interval, as well as the effective hourly arrival rate in each direction.*

Approach	Movement	Recorded	Hourly
R44	Right	28	112
	Left	1	4
Molteno	Right	0	0
	Left	39	156

TABLE 9.2: *Vehicle turn counts from the R44 into Molteno Road, and from Molteno Road into the R44, during a 15-minute interval, as well as the effective hourly arrival rate in each direction.*

Approach	Movement	Recorded	Hourly
R44	Right	9	36
	Left	10	40
Bell	Right	4	16
	Left	28	112

TABLE 9.3: *Vehicle turn counts from the R44 into Bell Road, and from Bell Road into the R44, during a 15-minute interval, as well as the effective hourly arrival rate in each direction.*

Each of the eight signalised intersections have either three or four green phases, depending on the nature of the intersection. Intersections 4, 5 and 6 in Figure 9.1 all have three traffic approaches and thus only require three green phases: A green phase for vehicles travelling northbound or southbound along the R44, secondly, a protected right-turn phase for vehicles travelling along the R44 and wishing to turn right onto the adjacent street, and finally a green phase for vehicles wishing to turn left or right onto the R44. Intersections 1, 3, 7, and 8 all have four traffic approaches and therefore four green phases. These phases include the aforementioned three phases (as for intersections 4, 5 and 6), with an additional phase that awards a protected right-turn phase for vehicles travelling along the side streets and wishing to turn right onto the R44. Intersection 2 is modelled slightly differently because, although it has four approaches, traffic volume turning right onto the R44 from the side streets is minor and so a protected right-turn phase is not necessary. As was the case in the models of §6, the exclusive right-turn phase is skipped if fewer than four vehicles are waiting to turn right.

It is also noted that right-turning by west-bound traffic at intersection 3 onto the R44 is not permitted. Similarly, south-bound traffic at intersection 3 are not permitted to turn right.

9.2 Simulation results in R44 model

The traffic signal phases, their orderings and the arrival rates (described in the previous section) are all taken as input data for the simulation model. The model was run for a single simulation hour between 6am and 7am, and again for a period of two simulation hours between 10am and 12pm. The relative performances of the algorithms were recorded over these specific time intervals in order to capture the state of the road network during very low traffic demands of the early morning, and moderate traffic experienced just before midday. The results of these simulations are reported and interpreted in the remainder of this section.

9.2.1 Results under very light traffic conditions

The ANOVA column in Table 9.4 indicates that for all six PMIs there are statistical differences between the means returned by the algorithms at a 5% level of significance. Furthermore, the outcome of the Levene test revealed that the algorithmic variances of the mean samples were statistically distinguishable for all six PMIs and therefore the Games-Howell *post hoc* test was used in order to determine between which pairs of algorithmic outputs differences are statistically discernible in respect of these six PMIs.

	Mean value								p-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA	ANOVA	Levene's Test
1	40.39	45.99	39.66	39.02	50.29	58.92	42.30	42.13	$<1 \times 10^{-17}$	6.27×10^{-4}
2	1.448	1.506	1.447	1.451	1.555	1.708	1.466	1.514	$<1 \times 10^{-17}$	5.30×10^{-14}
3	0.388	0.462	0.367	0.373	0.608	0.972	0.406	0.517	$<1 \times 10^{-17}$	1.51×10^{-9}
4	0.140	0.166	0.133	0.143	0.224	0.414	0.144	0.202	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
5	13.94	18.34	13.49	13.41	19.83	28.37	15.30	17.10	$<1 \times 10^{-17}$	1.07×10^{-6}
6	0.0992	0.1220	0.0982	0.1000	0.1338	0.1858	0.1054	0.1277	$<1 \times 10^{-17}$	1.21×10^{-4}

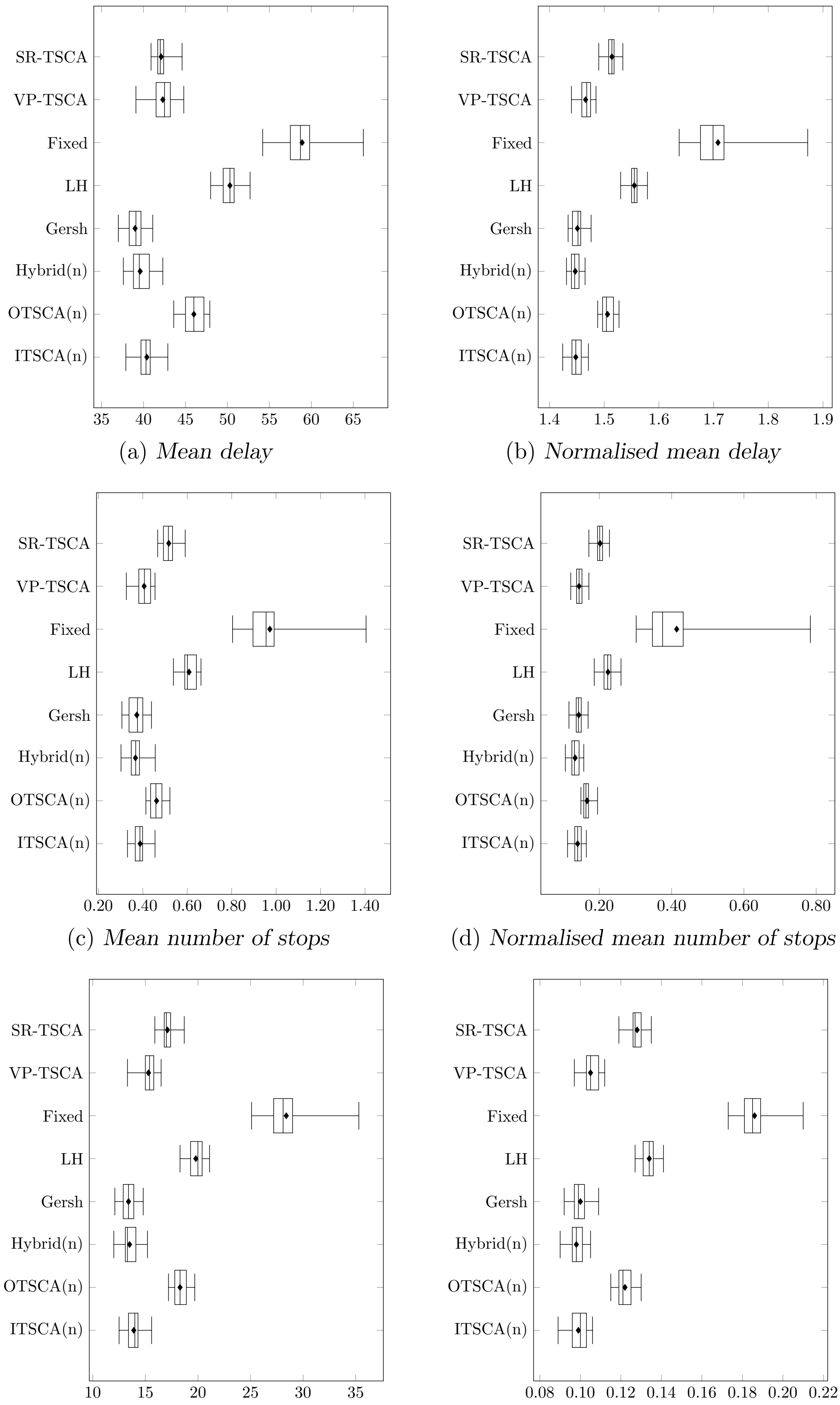
Table 9.4: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under very light traffic conditions along the R44 eight-intersection corridor. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

The best performances in respect of mean and normalised mean delay were returned by Gersh, Hybrid(n) and the I-TSCA(n), while none of these three algorithms differed significantly from one another. The VP-TSCA and the SR-TSCA achieved the next best mean delay values of 42.30 seconds and 42.13 seconds, respectively, while the VP-TSCA and the O-TSCA(n) achieved the next best normalised mean delay values of 1.466 and 1.448, respectively, although both pairs of algorithms obtained statistically indistinguishable values. LH was out performed by all other algorithms with the exception of Fixed, for both mean and normalised mean delay.

Once again Gersh, Hybrid(n) and the I-TSCA(n) achieved the most favourable results, for both mean and normalised mean number of stops, followed closely by VP-TSCA. The O-TSCA(n) is the next best algorithm in both these respects, achieving a normalised mean number of stops value of 0.166, indicating that vehicles under the control of the O-TSCA(n) are likely to stop at approximately 17% of the intersections they encounter. Once again a relatively poor performance was exhibited by LH and Fixed, as well as by the SR-TSCA, in respect of mean and normalised mean number of stops, which is further evidence of previous claims that these algorithms do not offer appropriate control under light traffic conditions.

Unsurprisingly, Gersh, Hybrid(n) and the I-TSCA(n) obtained the best results for the remaining two PMI values. The next best performing algorithm in both instances is the VP-TSCA, which achieved a mean and normalised mean time spent travelling under 10km/h value of 15.30 seconds and 0.1054. These values indicate that when the VP-TSCA is employed at signalised intersections, vehicles spend approximately 15 seconds of their journey travelling at very slow speeds, which amounts to roughly 13% of their total time spent in the road network.

Some of these results are not surprising, and corroborate statements reported in previous chapters. For instance it was discovered in the simulation results of Chapter 6 that the I-TSCA(n),



(a) *Mean delay*

(b) *Normalised mean delay*

(c) *Mean number of stops*

(d) *Normalised mean number of stops*

(e) *Mean time spent travelling under 10km/h*

(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 9.2: *PMI results for Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 within the context of the R44 eight-intersection corridor under very light traffic conditions.*

p-values of the Games-Howell test: Mean delay								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.48×10^{-11}	2.44×10^{-1}	4.90×10^{-4}	1.49×10^{-11}	1.11×10^{-12}	2.32×10^{-6}	7.70×10^{-7}
O-TSCA(n)		—	1.41×10^{-11}	1.28×10^{-11}	1.49×10^{-11}	1.13×10^{-12}	1.48×10^{-11}	$<1 \times 10^{-17}$
Hybrid(n)			—	3.51×10^{-1}	1.45×10^{-11}	9.02×10^{-13}	1.65×10^{-10}	1.19×10^{-11}
Gersh				—	1.35×10^{-11}	6.24×10^{-13}	1.17×10^{-11}	4.30×10^{-12}
LH					—	1.11×10^{-12}	1.45×10^{-11}	$<1 \times 10^{-17}$
Fixed						—	1.05×10^{-12}	$<1 \times 10^{-17}$
VP-TSCA							—	9.98×10^{-1}
SR-TSCA								—
Mean	40.39	45.99	39.66	39.02	50.29	58.92	42.30	42.13

TABLE 9.5: *Differences in respect of the mean delay returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under very lighter traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean delay								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.16×10^{-11}	9.99×10^{-1}	9.85×10^{-1}	1.00×10^{-11}	1.21×10^{-13}	1.03×10^{-5}	$<1 \times 10^{-17}$
O-TSCA(n)		—	9.53×10^{-12}	1.49×10^{-11}	1.48×10^{-11}	1.01×10^{-13}	1.48×10^{-11}	6.05×10^{-2}
Hybrid(n)			—	7.75×10^{-1}	1.12×10^{-11}	7.09×10^{-14}	6.56×10^{-8}	1.41×10^{-11}
Gersh				—	1.48×10^{-11}	1.11×10^{-13}	7.23×10^{-5}	5.43×10^{-12}
LH					—	1.18×10^{-13}	1.45×10^{-11}	6.97×10^{-12}
Fixed						—	1.05×10^{-13}	7.07×10^{-14}
VP-TSCA							—	3.23×10^{-12}
SR-TSCA								—
Mean	1.448	1.506	1.447	1.451	1.555	1.708	1.466	1.514

TABLE 9.6: *Differences in respect of the normalised mean delay returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under very light traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	6.17×10^{-11}	2.27×10^{-1}	7.26×10^{-1}	7.99×10^{-12}	1.10×10^{-13}	3.40×10^{-1}	1.43×10^{-11}
O-TSCA(n)		—	1.50×10^{-11}	1.90×10^{-11}	1.40×10^{-11}	5.78×10^{-14}	6.59×10^{-7}	7.88×10^{-7}
Hybrid(n)			—	9.98×10^{-1}	1.39×10^{-11}	6.08×10^{-14}	8.30×10^{-4}	1.41×10^{-11}
Gersh				—	1.40×10^{-11}	$<1 \times 10^{-17}$	1.64×10^{-2}	7.02×10^{-12}
LH					—	1.45×10^{-14}	1.32×10^{-11}	1.18×10^{-11}
Fixed						—	7.23×10^{-14}	8.62×10^{-14}
VP-TSCA							—	1.46×10^{-11}
SR-TSCA								—
Mean	0.388	0.462	0.367	0.373	0.608	0.972	0.406	0.517

TABLE 9.7: *Differences in respect of the mean number of vehicle stops returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under very light traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	4.83×10^{-9}	4.90×10^{-1}	9.93×10^{-1}	9.64×10^{-12}	2.98×10^{-12}	8.89×10^{-1}	1.37×10^{-11}
O-TSCA(n)		—	1.48×10^{-11}	6.85×10^{-8}	2.47×10^{-12}	3.50×10^{-11}	2.44×10^{-7}	9.12×10^{-12}
Hybrid(n)			—	1.13×10^{-1}	1.13×10^{-11}	1.56×10^{-12}	2.68×10^{-2}	1.44×10^{-11}
Gersh				—	7.47×10^{-12}	3.86×10^{-12}	9.99×10^{-1}	1.26×10^{-11}
LH					—	1.79×10^{-8}	3.22×10^{-12}	1.05×10^{-5}
Fixed						—	4.61×10^{-12}	1.49×10^{-9}
VP-TSCA							—	9.56×10^{-12}
SR-TSCA								—
Mean	0.140	0.166	0.133	0.143	0.224	0.414	0.144	0.202

TABLE 9.8: *Differences in respect of the normalised mean number of stops returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under very light traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.45×10^{-11}	2.45×10^{-1}	1.07×10^{-1}	1.48×10^{-11}	$<1 \times 10^{-17}$	5.98×10^{-8}	8.02×10^{-12}
O-TSCA(n)		—	1.48×10^{-11}	1.49×10^{-11}	1.72×10^{-9}	$<1 \times 10^{-17}$	1.47×10^{-11}	4.83×10^{-8}
Hybrid(n)			—	9.99×10^{-1}	1.44×10^{-11}	$<1 \times 10^{-17}$	1.66×10^{-11}	1.21×10^{-11}
Gersh				—	1.46×10^{-11}	$<1 \times 10^{-17}$	1.51×10^{-11}	1.16×10^{-11}
LH					—	$<1 \times 10^{-17}$	1.49×10^{-11}	9.20×10^{-12}
Fixed						—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
VP-TSCA							—	9.18×10^{-12}
SR-TSCA								—
Mean	13.94	18.34	13.49	13.41	19.83	28.37	15.30	17.10

TABLE 9.9: *Differences in respect of the mean time vehicles spent travelling under 10km/h returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under very light traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	1.12×10^{-11}	9.86×10^{-1}	9.98×10^{-1}	1.24×10^{-11}	7.59×10^{-13}	2.28×10^{-5}	$<1 \times 10^{-17}$
O-TSCA(n)		—	1.44×10^{-11}	1.47×10^{-11}	1.48×10^{-11}	8.10×10^{-13}	1.37×10^{-11}	2.57×10^{-6}
Hybrid(n)			—	7.26×10^{-1}	1.47×10^{-11}	1.07×10^{-12}	2.83×10^{-7}	4.42×10^{-12}
Gersh				—	1.49×10^{-11}	9.74×10^{-13}	9.28×10^{-5}	6.10×10^{-12}
LH					—	9.64×10^{-13}	1.4×10^{-11}	7.90×10^{-7}
Fixed						—	1.13×10^{-12}	$<1 \times 10^{-17}$
VP-TSCA							—	2.39×10^{-12}
SR-TSCA								—
Mean	0.0992	0.1220	0.0982	0.1000	0.1338	0.1858	0.1054	0.1277

TABLE 9.10: *Differences in respect of the normalised mean time vehicles spent travelling under 10km/h returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under very light traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

Hybrid(n) and Gersh were particularly effective under lighter traffic conditions (although not as light as in this scenario). Similarly, it was found that algorithms such as Fixed, LH and the SR-TSCA were more effective under heavier traffic conditions, which explains their poor performance here. It is, however, surprising that the VP-TSCA and the SR-TSCA achieved such similar results (particularly for mean delay), since these algorithms were shown in Chapter 7 to operate most effectively under contrasting traffic conditions.

9.2.2 Results under moderate traffic conditions

The ANOVA column in Table 9.11 indicates that for all six PMIs there are again statistical differences between the means returned by the various algorithms at a 5% level of significance. Furthermore, the outcome of the Levene test revealed that the algorithmic variances of the mean samples were statistically indistinguishable for PMI 6, and therefore the Fisher LSD *post hoc* test was used in order to determine between which pairs of algorithmic outputs differences are statistically discernible in respect of this particular PMI. The Games-Howell *post hoc* test was used for this purpose in respect of the remaining five PMIs, as the Levene test indicated that there was a statistical difference between the variances of the sample means for these PMIs at a 5% level of significance.

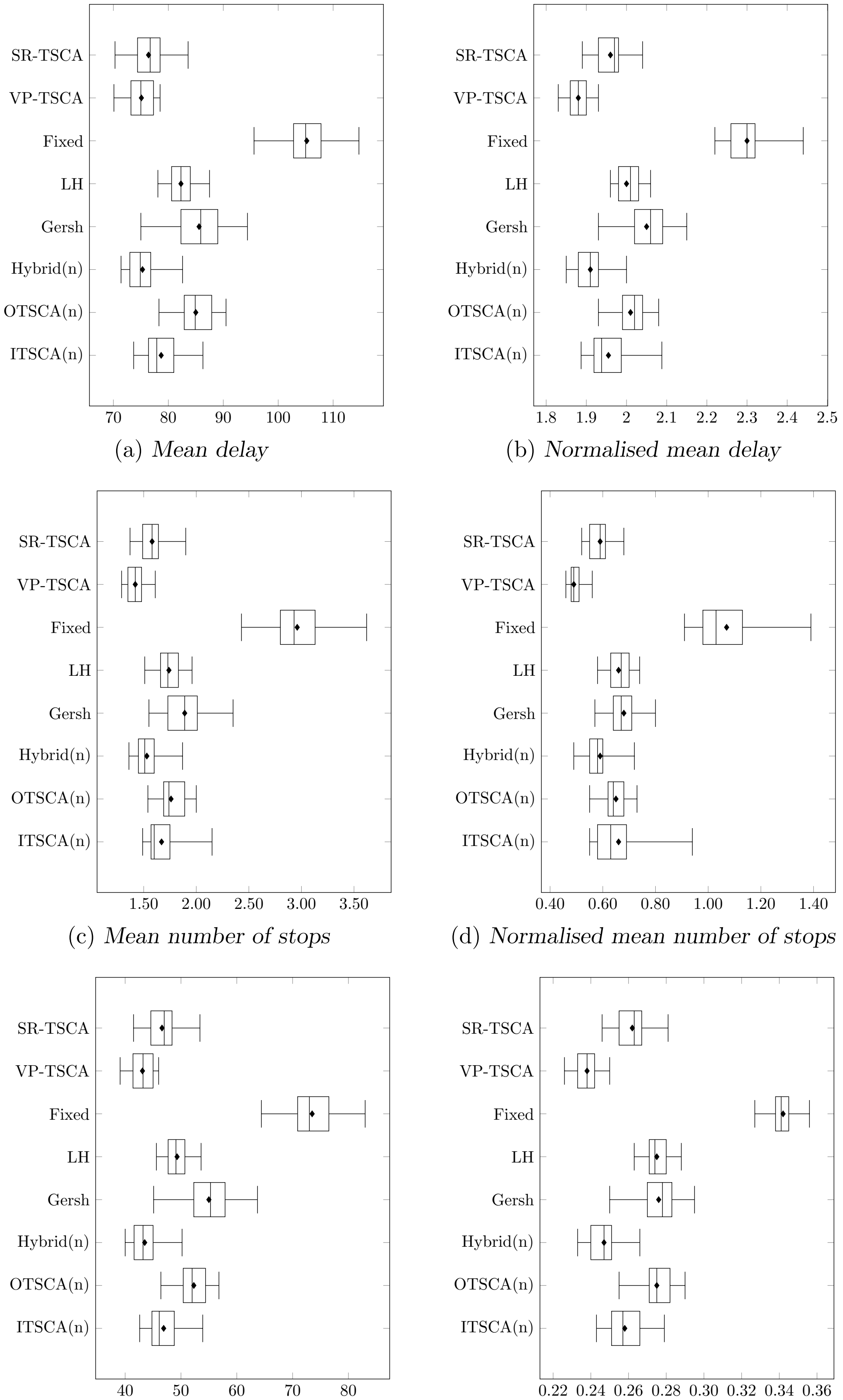
	Mean value								p-value	
PMI	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA	ANOVA	Levene's Test
1	78.71	84.97	75.28	85.61	82.32	105.18	75.06	76.42	$<1 \times 10^{-17}$	2.51×10^{-3}
2	1.956	2.015	1.909	2.055	2.005	2.302	1.878	1.961	$<1 \times 10^{-17}$	1.09×10^{-3}
3	1.671	1.760	1.529	1.887	1.740	2.961	1.415	1.578	$<1 \times 10^{-17}$	3.59×10^{-10}
4	0.655	0.647	0.586	0.678	0.664	1.074	0.495	0.590	$<1 \times 10^{-17}$	1.21×10^{-14}
5	46.93	52.27	43.48	55.01	49.32	73.46	43.13	46.62	$<1 \times 10^{-17}$	4.80×10^{-5}
6	0.2583	0.2753	0.2465	0.2765	0.2754	0.3423	0.2381	0.2623	$<1 \times 10^{-17}$	5.82×10^{-2}

TABLE 9.11: *The mean values of the six PMIs, as well as the p-values for the ANOVA and Levene statistical tests under moderate traffic conditions along the R44 eight-intersection corridor. PMI 1 and PMI 2 represent the mean and normalised mean delay experienced by vehicles in the road network, respectively. PMI 3 and PMI 4 are the mean and normalised mean number of stops, respectively, while PMI 5 and PMI 6 are the mean and normalised mean time vehicles spent travelling under 10km/h, respectively. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

The VP-TSCA was not outperformed by any other algorithm in respect of mean and normalised mean delay, achieving values of 75.06 seconds and 1.878, respectively. The VP-TSCA outperformed all other algorithms in respect of mean delay except for Hybrid(n) and the SR-TSCA from which it did not differ significantly, while significantly outperforming all seven algorithms in respect of normalised mean delay at a 5% level of significance. In terms of normalised mean delay, Hybrid(n) was the next best performing algorithm and achieved a value of 1.909, indicating that vehicles spend, on average, an additional 91% of their time in the road network as a result of delays.

A similar ranking was apparent in respect of the mean and normalised mean number of stops. The VP-TSCA significantly outperformed all the other algorithms with respect to these two PMIs, followed by Hybrid(n) and the SR-TSCA, whose performances did not differ significantly from one another. The I-TSCA(n), the O-TSCA(n), Gersh and LH all achieved very similar results for these two PMIs, not differing significantly from one another. Fixed returned the poorest results, with vehicles stopping approximately twice as many times when compared with the other seven algorithms.

The VP-TSCA outperformed all other algorithms except for Hybrid(n), from which it did not differ significantly, in respect of mean time spent travelling under 10km/h. The I-TSCA(n) and



(a) *Mean delay*

(b) *Normalised mean delay*

(c) *Mean number of stops*

(d) *Normalised mean number of stops*

(e) *Mean time spent travelling under 10km/h*

(f) *Normalised mean time spent travelling under 10km/h*

FIGURE 9.3: *PMI results for Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 within the context of the R44 eight-intersection corridor under moderate traffic conditions.*

p-values of the Games-Howell test: Mean delay								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	7.88×10^{-8}	2.24×10^{-3}	3.73×10^{-7}	8.62×10^{-4}	4.34×10^{-12}	4.23×10^{-4}	1.79×10^{-1}
O-TSCA(n)		—	5.36×10^{-12}	9.98×10^{-1}	2.58×10^{-2}	1.23×10^{-12}	$<1 \times 10^{-17}$	1.64×10^{-11}
Hybrid(n)			—	$<1 \times 10^{-17}$	1.48×10^{-11}	$<1 \times 10^{-17}$	9.99×10^{-1}	8.32×10^{-1}
Gersh				—	2.13×10^{-2}	1.49×10^{-11}	1.27×10^{-12}	5.70×10^{-11}
LH					—	$<1 \times 10^{-17}$	1.18×10^{-11}	8.67×10^{-9}
Fixed						—	1.02×10^{-12}	$<1 \times 10^{-12}$
VP-TSCA							—	5.96×10^{-1}
SR-TSCA								—
Mean	78.71	84.97	75.28	85.61	82.32	105.18	75.06	76.42

Table 9.12: *Differences in respect of the mean delay returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under moderate traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean delay								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	3.53×10^{-4}	4.53×10^{-3}	3.16×10^{-8}	1.73×10^{-3}	1.47×10^{-11}	2.97×10^{-7}	9.99×10^{-1}
O-TSCA(n)		—	6.17×10^{-12}	3.19×10^{-12}	9.64×10^{-1}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	7.90×10^{-5}
Hybrid(n)			—	$<1 \times 10^{-17}$	1.31×10^{-11}	$<1 \times 10^{-17}$	3.36×10^{-3}	2.31×10^{-5}
Gersh				—	9.67×10^{-4}	1.40×10^{-11}	6.65×10^{-13}	4.61×10^{-9}
LH					—	4.37×10^{-13}	3.26×10^{-12}	2.90×10^{-4}
Fixed						—	1.69×10^{-13}	$<1 \times 10^{-17}$
VP-TSCA							—	1.13×10^{-11}
SR-TSCA								—
Mean	1.956	2.015	1.909	2.055	2.005	2.302	1.878	1.961

Table 9.13: *Differences in respect of the normalised mean delay returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under moderate traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	3.68×10^{-1}	1.07×10^{-2}	9.07×10^{-4}	6.47×10^{-1}	$<1 \times 10^{-17}$	2.53×10^{-7}	2.80×10^{-1}
O-TSCA(n)		—	1.88×10^{-8}	8.06×10^{-2}	9.99×10^{-1}	5.44×10^{-13}	$<1 \times 10^{-17}$	1.96×10^{-5}
Hybrid(n)			—	7.80×10^{-10}	5.18×10^{-8}	$<1 \times 10^{-17}$	6.70×10^{-4}	7.26×10^{-1}
Gersh				—	2.07×10^{-2}	$<1 \times 10^{-17}$	2.61×10^{-13}	6.09×10^{-8}
LH					—	9.10×10^{-15}	$<1 \times 10^{-17}$	7.77×10^{-5}
Fixed						—	3.50×10^{-14}	9.62×10^{-14}
VP-TSCA							—	4.35×10^{-6}
SR-TSCA								—
Mean	1.671	1.760	1.529	1.887	0.740	2.96	1.415	1.578

TABLE 9.14: *Differences in respect of the mean number of vehicle stops returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under moderate traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Normalised mean number of stops								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	9.99×10^{-1}	3.49×10^{-2}	9.64×10^{-1}	9.99×10^{-1}	$<1 \times 10^{-17}$	3.55×10^{-8}	4.57×10^{-2}
O-TSCA(n)		—	3.05×10^{-4}	3.02×10^{-1}	8.26×10^{-1}	$<1 \times 10^{-17}$	1.10×10^{-12}	1.49×10^{-4}
Hybrid(n)			—	3.76×10^{-7}	1.25×10^{-6}	$<1 \times 10^{-17}$	1.86×10^{-9}	9.99×10^{-1}
Gersh				—	9.61×10^{-1}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	1.61×10^{-7}
LH					—	$<1 \times 10^{-17}$	8.46×10^{-13}	2.13×10^{-7}
Fixed						—	6.76×10^{-14}	$<1 \times 10^{-17}$
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	0.655	0.647	0.586	0.678	0.664	1.074	0.495	0.590

TABLE 9.15: *Differences in respect of the normalised mean number of stops returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under moderate traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Games-Howell test: Mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	3.32×10^{-7}	4.73×10^{-4}	5.16×10^{-10}	3.13×10^{-2}	$<1 \times 10^{-17}$	3.53×10^{-5}	9.99×10^{-1}
O-TSCA(n)		—	4.52×10^{-12}	8.41×10^{-2}	1.58×10^{-3}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	1.64×10^{-8}
Hybrid(n)			—	$<1 \times 10^{-17}$	1.58×10^{-11}	4.75×10^{-13}	9.98×10^{-1}	7.13×10^{-4}
Gersh				—	1.03×10^{-6}	1.42×10^{-11}	7.67×10^{-13}	6.84×10^{-11}
LH					—	1.11×10^{-12}	1.02×10^{-11}	3.93×10^{-3}
Fixed						—	3.13×10^{-13}	$<1 \times 10^{-17}$
VP-TSCA							—	3.73×10^{-5}
SR-TSCA								—
Mean	46.93	52.27	43.48	55.01	49.32	73.46	43.13	46.62

TABLE 9.16: *Differences in respect of the mean time vehicles spent travelling under 10km/h returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under moderate traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

p-values of the Fisher LSD test: Normalised mean time spent travelling under 10km/h								
Algorithm	I-TSCA(n)	O-TSCA(n)	Hybrid(n)	Gersh	LH	Fixed	VP-TSCA	SR-TSCA
I-TSCA(n)	—	2.29×10^{-13}	1.46×10^{-7}	6.55×10^{-15}	1.80×10^{-13}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	6.66×10^{-2}
O-TSCA(n)		—	$<1 \times 10^{-17}$	5.78×10^{-1}	9.96×10^{-1}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	1.02×10^{-8}
Hybrid(n)			—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	1.29×10^{-4}	5.56×10^{-12}
Gersh				—	6.05×10^{-1}	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	4.89×10^{-10}
LH					—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$	8.32×10^{-9}
Fixed						—	$<1 \times 10^{-17}$	$<1 \times 10^{-17}$
VP-TSCA							—	$<1 \times 10^{-17}$
SR-TSCA								—
Mean	0.2583	0.2753	0.2465	0.2765	0.2754	0.3423	0.2381	0.2623

TABLE 9.17: *Differences in respect of the normalised mean time vehicles spent travelling under 10km/h returned by Fixed and the seven self-organising algorithms of §2.1.1, §2.1.2, §5.3, §7.1 and §7.2 under moderate traffic conditions within the context of the R44 eight-intersection corridor. A table entry less than* 0.05 *(indicated in red) denotes a difference at a 5% level of significance.*

the SR-TSCA achieved the next best results in terms of this measure, followed by LH, which is ranked fifth. The VP-TSCA outperformed all seven algorithms in respect of the normalised measure of time vehicles spend travelling slowly, achieving a value of 0.2381, implying that vehicles spend approximately 24% of their time in the road network travelling a maximum speed of 10km/h. Hybrid(n) obtained the next best result with a value of 0.2465. The I-TSCA(n) achieved the third best result in terms of this measure, followed by the O-TSCA(n), Gersh and LH, whose performances did not differ significantly from one another.

The VP-TSCA was undoubtedly the best performing algorithm overall under moderate traffic conditions as it was not outperformed by any other algorithm for any of the six PMIs. The VP-TSCA has therefore been shown to perform relatively well under very light, light and moderate traffic conditions, indicating that the performance of this algorithm is not particularly dependant on the current traffic conditions. This is attributed to the hyperbolic function relating the road saturation to the inter-vehicle threshold distances associated with the VP-TSCA logic — adapting its behaviour based on the current level of traffic congestion. Hybrid(n) is the next best algorithm under moderate traffic conditions, performing on par with the VP-TSCA for two of the six PMIs, and ranking second best for the remaining four PMIs. The results returned by the SR-TSCA and the I-TSCA(n) are not distinguishable from one another for any of the six PMIs at a 5% level of significance, and are the next best performing algorithms. Fixed returned the worst results for all six PMIs, which may be surprising as it is better suited for heavier traffic conditions. This poor performance is largely attributed to the fact that the spacing between consecutive intersections is not uniform in the road corridor, as well as that traffic flow densities at various intersections vary. This hinders the formation of green waves through consecutive intersections, resulting in an increased number of vehicle stops.

9.3 Chapter summary

The chapter opened in §9.1 with a description of the road corridor considered as a case study in this chapter, namely an arterial road referred to as the R44, which consists of eight signalised intersections and three stop-sign controlled approaching side streets. The results of the algorithmic comparison for this corridor under very light traffic conditions were reported in §9.2.1. It was found that the I-TSCA(n), Hybrid(n) and Gersh were the best performing algorithms, while Fixed, LH and the SR-TSCA did not perform effectively. The algorithms were then compared in §9.2.2 under moderate traffic conditions along the same corridor, and it was found that the VP-TSCA obtained the best results, followed by Hybrid(n). Fixed, on the other hand, performed poorly for all six PMIs under moderate traffic conditions.

CHAPTER 10

Conclusion

Contents

This chapter contains a summary of the work presented in this dissertation (in §10.1) and an appraisal of the contributions made in the dissertation (in §10.2). The chapter closes in §10.3 with a number of suggestions for possible, follow-up future work.

10.1 Dissertation summary

The dissertation opened in Chapter 1 with a brief discussion on the problems associated with traffic congestion and the various forms of traffic control strategies that are used to combat these problems. The notions of self-organisation and emergence were reviewed as facilitating a promising approach towards traffic signal control. An informal description of the problem considered in this dissertation was also given, together with the scope and objectives to be pursued during the research documented in this dissertation.

An extensive literature review on seven self-organising traffic signal control algorithms was presented in Chapter 2, in fulfilment of Objectives I(a) and I(b) of §1.3. The chapter opened with a section on three self-organising algorithms proposed in 2012 and earlier. The first algorithm was proposed by Gershenson and Rosenblueth [22], and is a rule-based algorithm that switches signals once certain conditions are met. The second algorithm was proposed by Lämmer and Helbing [46], and was inspired by the observation of the natural occurrence of self-organisation found in oscillations of pedestrian flows through narrow bottlenecks. The final early self-organising algorithm reviewed in this section was an algorithm proposed by Xie *et al.* [87], in which similar vehicle flows are aggregated into clusters that are considered in their entireties at distinct decision points in time. In the following section, three recently proposed self-organising traffic signal control algorithms proposed by two PhD students were reviewed. These include an algorithm put forward by Cesme [12], which aims to alleviate traffic pressure along arterial roads, as well as three algorithms proposed by Einhorn [16]. The first of these was an algorithm inspired the theory of inventory control, considering the virtual costs associated with delay time experienced by vehicles. The second algorithm was inspired by the chemical process of osmosis. This algorithm draws parallels between the push and pull forces experienced between solvent and solute molecules with the virtual forces experienced in traffic control. The third algorithm by Einhorn [16] is a hybrid algorithm that utilises both of the two aforementioned algorithms in order to exploit the best characteristics of both. This chapter closed with an identification of five of the seven algorithms that were to be compared later in the dissertation.

In pursuit of Objective I(c), Chapter 3 was dedicated to a thorough review of the components considered important when building a successful simulation model. The chapter opened with definitions of terms commonly associated with traffic control, and this was followed by brief descriptions of the four existing types of simulation modelling paradigms. The benefits and shortcomings of simulation modelling were discussed as were the typical steps followed in a simulation study. A number of model verification and validation techniques were also reviewed. The chapter closed with a review of various types of traffic simulation models, together with suitable software packages for each of these.

The newly designed and implemented microscopic traffic simulation model employed as a test bed in this dissertation was introduced in Chapter 4 in fulfilment of Objective II. The chapter opened with a detailed description of the modelling framework, including the construction of the road network embedded in the model, the implementation of traffic signals in the model, the generation of vehicles in the model and the nature of the simulation model output. The verification and validation techniques performed in respect of the simulation model were mentioned, with a strong focus on model validation (in fulfilment of Objective III). This was achieved by the replication of a real-world signalised intersection within the simulation model, with known arrival rates, turning probabilities and green times. The chapter closed with a description of the experimental design according to which experiments were to be carried out later in the dissertation. The design included choosing an appropriate length of simulation warm-up period, general specifications of the road networks to be considered and the vehicles travelling through it, and finally, the kinds of statistical tests to be performed in respect of the output data returned by the simulation model.

Chapter 5 opened with detailed descriptions of the logic behind a fixed-time control strategy and each of the five self-organising algorithms identified in Chapter 2 for simulation comparison purposes, as well as how they were implemented within the simulation model of Chapter 4 . These algorithms were subsequently compared in terms of six PMIs and these results were presented and interpreted under both lighter and heavier traffic conditions. Three algorithms exhibited relatively poor performance and so a number of changes were suggested for these algorithms and implemented in pursuit of Objectives V and VI. These algorithmic changes were found to result in a significant improvement in respect of the aforementioned PMIs and thus Objective VII was achieved.

Chapter 6 served in partial fulfilment of Objective IV, and contained a description of the implementation and comparison of the algorithms in the two road network topologies considered by Einhorn [16]. The first topology considered was a four-intersection corridor and it was found that under heavy traffic conditions Fixed and the O-TSCA(n) obtained the best algorithmic results, while none of the algorithms performed significantly better than the others under light traffic conditions. The second topology considered was a 3×4 grid of intersections and it was found that under light traffic conditions the O-TCSA(n), Gersh and Hybrid(n) obtained slightly better results than the other algorithms, while LH and Fixed were found to be the superior algorithms over all under heavy traffic conditions.

Two new self-organising algorithms were proposed in Chapter 7 in fulfilment of Objective VIII. The first of these algorithms is referred to the VP-TSCA and clusters approaching vehicles into groups depending on their distances from one another, while the second algorithm (referred to as the SR-TSCA) is concerned with the ratio of road saturation of competing traffic flows. These two algorithms were compared to the existing algorithms in both a four-intersection corridor as well as in a 3×4 grid of intersections under both light and heavy traffic conditions. The results returned by the algorithms in the corridor road network suggested that the VP-TSCA and Hybrid(n) were best suited for use under light traffic conditions, while Fixed and VP-TSCA achieved the best results under heavy traffic conditions. In the context of a grid road network, on the other hand, the VP-TSCA and Gersh achieved the best performances under light traffic conditions, while Fixed achieved the most favourable results under heavy traffic conditions, followed by the SR-TSCA.

All eight algorithms were implemented in a wide variety of traffic scenarios and statistically compared in Chapter 8, in fulfilment of Objective IV. The first two scenarios were carried out in the context of a 3×4 grid of intersections, in which one of the roads were closed and the arrival rate of the vehicles was varied, respectively. The comparisons carried out in these two scenarios were aimed at testing the abilities of the various algorithms to overcome a network disruption and deal with abnormal traffic conditions. The following two scenarios involved a corridor of intersections. The first of these scenarios consisted of a comparison of the relative performances of the algorithms in four, six and eight-intersection corridors in order to observe how the algorithms scaled with an increased network size. The final scenario involved six-intersection corridors in which the distances between consecutive intersections were varied. This tested how the reliant the performances of the algorithms were on the spacings between intersections.

In Chapter 9, the eight algorithms were compared in a real-world road network model making use of real traffic data, in fulfilment of Objective IX. The comparison took place during two different periods during the day, namely early morning (very light traffic conditions) and late morning (average traffic conditions). During the early morning time period it was found that Gersh and Hybrid(n) performed the best overall. The flexible nature of these two algorithms allowed them to switch signals frequently in order to serve the sparsely located vehicles. The VP-TSCA achieved the best results during the late morning time period, achieving a relatively good balance between flexibility and coordination over consecutive intersections.

10.2 Appraisal of dissertation contributions

The main contributions of this dissertation are sixfold. A brief summary and appraisal of these contributions are given in this section.

Contribution 1 *A microscopic traffic simulation modelling framework was developed in Anylogic which realistically represents vehicles travelling through a road network.*

The simulation modelling framework of Chapter 4 incorporates individual vehicle attributes such as length, preferred speed, acceleration and deceleration. Vehicle turns and lane changing are also supported, allowing vehicles to transition into the correct lane ahead of an upcoming turn, while specific vehicle routes may be specified by the user. The framework is customisable in the sense that the user may easily adjust parameters of vehicles such as arrival rates, vehicle attributes and turning probabilities. Various elements of the road network itself may also easily be adjusted such as altering the number of lanes contained in a road, the specification of left-hand or right-hand driving as well as the general appearance of the network. The framework supports the implementation of multiple traffic signal control algorithms that assume the use of radar detection. A range of different road network topologies may easily be implemented in the simulation modelling framework. Simulation replication visualisations may be observed during simulation model runs, and various data pertaining to the performance of the algorithms may be monitored throughout a simulation run.

Contribution 2 *The introduction of two novel performance measure indicators that were used to measure the relative performance of self-organising algorithms.*

During the statistical analysis of the relative performances of traffic signal control algorithms, four of the PMIs proposed by Einhorn [16] were utilised in this dissertation, namely the mean delay, the normalised mean delay, the mean number of stops and the normalised mean number of stops. It was noted, however, that these PMIs do not explicitly take into account vehicles that travel at slow speeds. Consider, for example, two vehicles that achieve the same PMI values for the four previously mentioned PMIs. It is possible that these two vehicles may achieve very different speeds throughout their routes through the road network, with one of the vehicles travelling at its preferred speed along most of its route and slowing down dramatically for the last part of the route, while the other vehicle may have travelled at a slightly slower speed than desired (due to slower vehicles ahead of it) and still obtain the same PMI values.

This deficiency was overcome with the introduction of two additional PMIs, namely the mean time spent travelling under 10km/h and the normalised mean time spent travelling under 10km/h. These PMIs record the amount of time vehicles spend travelling very slowly, although not coming to a complete stop and hence measure a different kind of frustration experienced by drivers if traffic conditions are poor. The normalised mean time spent travelling under 10km/h gives an indication of the percentage of time of their trips vehicles spend travelling under 10km/h.

Contribution 3 *Statistically significant improvements upon three self-organising algorithms from the literature, namely the O-TSCA, the I-TSCA and the Hybrid algorithm.*

While the three algorithms proposed by Einhorn [16] appeared promising, they underperformed during the initial comparison among themselves and with other self-organising algorithms from the literature. It was apparent that the I-TSCA switched signals too frequently so as to effectively allocate enough green time, while the O-TSCA was relatively inflexible in terms of switching signals, often assigning green times that were longer than necessary. Both these algorithms were therefore altered in order to rectify these shortcomings, and so was the Hybrid algorithm which is based on both the I-TSCA and O-TSCA. The three altered algorithms and their original counterparts were compared within the simulation framework of Chapter 4. The results revealed that all three of the original algorithms were significantly outperformed by their altered counterparts — and by a considerable margin — under both lighter and heavier traffic conditions.

Contribution 4 *The introduction of two novel self-organising traffic signal control algorithms.*

Through the observation of numerous simulation runs, a number of shortcomings became apparent in the logic behind some of the self-organising algorithms in the literature. Two new self-organising algorithms (referred to as the VP-TSCA and the SR-TSCA) were therefore proposed in which some of these shortcomings were addressed. In particular, it was found that algorithms based on real-time traffic data were generally more successful in terms of obtaining low PMI values than those that made use of predicted traffic data, and therefore both new algorithms utilised real-time data. The allocation of very short green times was also found to be inefficient in the existing algorithms and thus a minimum green time was employed for the SR-TSCA. Both of these novel algorithms returned favourable results under certain road network and traffic conditions.

Contribution 5 *A statistical comparison of the relative performance of a fixed-time control strategy and seven self-organising algorithms in a variety of hypothetical scenarios in respect of their propensities to facilitate green waves and recover from disruptions in different transportation network configurations.*

It was found that self-organising traffic signal control algorithms in the literature are typically evaluated in one or two standard road networks. In this dissertation, the algorithms were compared to one another in the context of a range of different road network topologies and traffic scenarios. These included a grid network under both lighter and heavier traffic conditions, in which commonly occurring road network disruptions were implemented, such as road closures and sudden increases of vehicle arrivals into the network. A corridor network was also considered, in which different numbers of intersections were present as well as three different configurations of intersection spacing. The comparisons were based on statistical analyses with respect to the model output data and conclusions were drawn from these analyses in terms of the most effective algorithms in each instance at a 95% level of confidence.

Contribution 6 *A statistical comparison of the relative performances of a fixed-time control strategy and seven self-organising algorithms within the context of a real road network.*

While the eight traffic signal control algorithms considered in this dissertation had already been compared in a variety of realistic scenarios, these scenarios were purely hypothetical. The fixed-time control strategy and the seven self-organising algorithms were therefore also finally compared in a real road network model in order to ascertain their relative performances in the context of a real-world scenario. This scenario accurately represented a corridor of eight intersections in Stellenbosch, South Africa and made use of real vehicle arrival rate data. Under very light traffic conditions it was found that Gersh achieved the best results, while the SR-TSCA obtained the best PMI values overall during moderate traffic conditions.

10.3 Future work

This section contains a number of suggestions for possible ways in which the research carried out in this dissertation may be extended.

10.3.1 Extend the scaling scenario

The scaling potential of the various algorithms was considered in §8.2.1 as a function of increasing corridor size in order to determine how the performance of the algorithms is affected by an increase in the complexity of the road network. It was found that the Hybrid(n), Gersh and Fixed were especially sensitive to an increase in the number of corridor intersections, while O-TSCA(n) and LH scaled very well (the PMI values of these algorithms scaled linearly in relation to the number of intersections). Finally, the I-TSCA(n), the VP-TSCA and the SR-TSCA incurred mild PMI increases between the four- to six-intersection corridor and the six- to eight-intersection corridor.

It is therefore suggested that the results for a ten and twelve-intersection corridor also be recorded for the algorithms, in order to gain a better idea of the relationship that exists between the PMIs and the number of intersections in respect of each algorithm. A similar test is suggested for grid road networks, by considering grids of different sizes. Since larger road networks require a substantially longer computation time, it is recommended that the algorithms be implemented in the context of 2×2, 3×3, 4×4 and 5×5 grids.

10.3.2 Test additional scenarios

While the self-organising algorithms were evaluated in a variety of different scenarios in this dissertation, there are still a number of other road network complexities that should be investigated. One of these involve the incorporation of pedestrians into the road network, as well as exclusive pedestrian phases into traffic signal cycles. These phases are common in signalised intersections and should therefore be included in the simulation model in order to improve its realism. The presence of pedestrians would be detected through the pushing of a button at an intersection, and so the number of pedestrians may not be detectable as is the case for vehicles. Pedestrian phases should therefore continue for a predetermined, fixed amount of time. Another scenario that should be considered is the inclusion of vehicle breakdowns and collisions, as it is unrealistic to assume that these do not occur. Finally, a scenario in which a large number of vehicles suddenly leave a particular point in the network should be investigated in order to ascertain whether the algorithmic results differ from those in §8.1.2 where a large number of vehicles arrived at a certain point in the road network.

10.3.3 Improve upon the I-TSCA(n)

While the I-TSCA(n) of §5.3.2 already dramatically improves upon the I-TSCA algorithm proposed by Einhorn [16], there is still further room for improvement in respect of this algorithm. The I-TSCA and I-TSCA(n) both calculate competing "costs" based on the required amount of green time that each phase would incur. This would be the ideal choice if the required green times calculated by the algorithms were optimal. This is generally not the case, however, as the required green time is calculated in respect of the time taken for the shortest queue to clear in the case of the I-TSCA and a similar calculation is carried out by the I-TSCA(n), as described in §5.3.2. It is therefore suggested that instead of calculating one specific required green time for each phase, the delay costs be calculated over a range of potential green time values. The phase and corresponding green time value that results in the lowest total cost may then be awarded service.

10.3.4 A hyper algorithm

It was found in §8.2.2 that Gersh performed better when the spacing between consecutive intersections is small, while the opposite was found for other algorithms such as Hybrid(n), which performed significantly worse when intersections were closely spaced to one another. Similarly, Gersh was found to perform extremely well under lighter traffic conditions and poorly under heavier traffic conditions, while the opposite was the case for LH. It is therefore suggested that combinations of algorithms be used for certain road networks and traffic conditions in order to exploit the most favourable characteristics of each algorithm. This may be achieved by employing a master or hyper algorithm which decides which of a number of slave algorithms, such as Gersh and LH, should be awarded control under different traffic conditions.

10.3.5 Measuring the propagation of green waves

Successful propagation of green waves as vehicles travel through a number of consecutive intersections unimpeded, indicates that a substantial level of coordination has been achieved between intersections. This coordination is partially recorded in the number of stops that vehicles are subjected to, since a vehicle that is rarely (if at all) required to stop, is likely to form part of a green wave. This does not, however, indicate which vehicles formed the green waves, nor does it suggest resulting green time offsets or the size (in vehicles) of the green waves. It is therefore suggested that the green times, offsets and the size of the platoon of vehicles that form the green wave during the green time be recorded. This will allow the coordination of intersections to be measured more accurately and the algorithms to be compared more effectively with one another.

References

[1] Altiok T & Melamed B, 2010, *Simulation modeling and analysis with Arena*, Academic Press, Amsterdam.

[2] Anylogic, 2016, *Dynamic systems modelling*, [Online], [Cited April 8$^{\text{th}}$, 2016], Available from `http://www.anylogic.com/anylogic/help/index.jsp?nav=/0_1_1`.

[3] AnyLogic, 2016, *Multimethod simulation software*, [Online], [Cited May 30$^{\text{th}}$, 2016], Available from `http://www.anylogic.com`.

[4] Balci O, 1998, *Verification, validation, and testing*, Proceedings of the 30$^{\text{th}}$ Winter Simulation Conference, Washington (DC), pp. 335–393.

[5] Banks J, 1998, *Handbook of simulation: Principles, methodology, advances, applications, and practice*, John Wiley & Sons, Atlanta (GA).

[6] Banks J, 2000, *Introduction to simulation*, Proceedings of the 32$^{\text{th}}$ Winter Simulation Conference, Orlando (FL), pp. 9–16.

[7] Behrisch M, Bieker L, Erdmann J & Krajzewicz D, 2011, *SUMO—simulation of urban mobility: An overview*, Proceedings of the 3rd International Conference on Advances in System Simulation, Barcelona, pp. 55–60.

[8] Bonabeau E, 2002, *Agent-based modeling: Methods and techniques for simulating human systems*, Proceedings of the National Academy of Sciences, pp. 7280–7287.

[9] Borshchev A & Filippov A, 2004, *From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools*, Proceedings of the 22nd International Conference of the System Dynamics Society, Oxford, No page numbers.

[10] Boxill SA & Yu L, 2000, *An evaluation of traffic simulation models for supporting ITS*, (Unpublished) Technical Report, Center for Transportation Training and Research, Houston (TX).

[11] Burghout W, Koutsopoulos HN & Andreasson I, 2006, *A discrete-event mesoscopic traffic simulation model for hybrid traffic simulation*, Proceedings of the 9$^{\text{th}}$ Intelligent Transportation Systems Conference, Toronto, pp. 1102–1107.

[12] Cesme B, 2013, *Self-organizing traffic signals for arterial control*, PhD thesis, Northeastern University, Boston (MA).

[13] Christopherson P & Riddle R, 1979, *Ideal street spacing tables for balanced progression*, (Unpublished) Technical Report, National Technical Information Service, Alexandria (VA).

[14] Cools SB, Gershenson C & D'Hooghe B, 2008, *Self-organizing traffic lights: A realistic simulation*, 2$^{\text{nd}}$ Edition, Springer, London.

[15] De Wolf T, Samaey G & Holvoet T, 2005, *Engineering self-organising emergent systems with simulation-based scientific analysis*, Proceedings of the 3^{rd} International Workshop on Engineering Self-Organising Applications, New York (NY), pp. 146–160.

[16] Einhorn MD, 2015, *Self-organising traffic control inspired by inventory theory and the process of osmosis*, PhD thesis, Stellenbosch University, Stellenbosch.

[17] Ferreira M, Fernandes R, Conceição H, Viriyasitavat W & Tonguz O, 2010, *Self-organized traffic control*, Proceedings of the 7^{th} ACM international workshop on Vehicular InterNETworking, Chicago (IL), pp. 85–90.

[18] Field A, 2000, *Contrasts and post hoc tests for one-way independent ANOVA using SPSS*, [Online], [Cited October 23$^{\text{rd}}$, 2016], Available from `http://www.statisticshell.com/docs/contrasts.pdf`.

[19] Games PA & Howell JF, 1976, *Pairwise multiple comparison procedures with unequal n's and/or variances: A Monte Carlo study*, Journal of Educational and Behavioral Statistics, **1(2)**, pp. 113–125.

[20] Geroliminis N & Daganzo CF, 2007, *Macroscopic modeling of traffic in cities*, Proceedings of the 86$^{\text{th}}$ Transportation Research Board Annual Meeting, Washington (DC), No. 07-0413.

[21] Gershenson C, 2005, *Self-organizing traffic lights*, Complex Systems, **16(1)**, pp. 29–53.

[22] Gershenson C & Rosenblueth DA, 2012, *Adaptive self-organization vs static optimization: A qualitative comparison in traffic light coordination*, Kybernetes, **41(3)**, pp. 386–403.

[23] Gogg TJ & Mott JRA, 1992, *Improve quality and productivity with simulation*, 3$^{\text{rd}}$ Edition, JMI Consulting Group, Lansing (MI).

[24] Goodwin P, 2004, *The economic costs of road traffic congestion*, The Rail Freight Group, University College London, London.

[25] Google, 2016, *Google Earth*, [Online], [Cited November 8$^{\text{th}}$, 2016], Available from `https://www.google.co.za/maps/`.

[26] Gordon RL & Tighe W, 2005, *Traffic control systems handbook*, (Unpublished) Technical Report, Dunn Engineering Associates, New York (NY).

[27] Hayter AJ, 1986, *The maximum familywise error rate of Fisher's least significant difference test*, Journal of the American Statistical Association, **81(396)**, pp. 1000–1004.

[28] Helbing D, Hennecke A, Shvetsov V & Treiber M, 2001, *MASTER: Macroscopic traffic simulation based on a gas-kinetic, non-local traffic model*, Transportation Research Part B: Methodological, **35(2)**, pp. 183–211.

[29] Heylighen F, 1989, *Self-organization, emergence and the architecture of complexity*, Proceedings of the 1^{st} European Conference on System Science, Stuttgart, pp. 23–32.

[30] Heylighen F, 2001, *The science of self-organization and adaptivity*, The Encyclopedia of Life Support Systems, **5(3)**, pp. 253–280.

[31] Hidas P, 2002, *Modelling lane changing and merging in microscopic traffic simulation*, Transportation Research Part C: Emerging Technologies, **10(5)**, pp. 351–371.

[32] *Highway capacity manual*, Transportation and Research Board, Washington (DC), 2000.

[33] Hilton A & Armstrong R, 2010, *One-way analysis of variance*, John Wiley & Sons, Inc., Hoboken (NJ).

[34] Howell JF & Games PA, 1973, *The robustness of the analysis of variance and the Tukey WSD test under various patterns of heterogeneous variances*, The Journal of Experimental Education, **41(4)**, pp. 33–37.

[35] Howell JF & Games PA, 1974, *The effects of variance heterogeneity on simultaneous multiple-comparison procedures with equal sample size*, British Journal of Mathematical and Statistical Psychology, **27(1)**, pp. 72–81.

[36] Huper J, 2009, *Macroscopic modeling and simulation of freeway traffic flow*, Proceedings of the 12$^{\text{th}}$ IFAC Symposium on Control in Transportation Systems, Los Angeles (CA), pp. 112–116.

[37] Ingalls RG, 2008, *Introduction to simulation*, Proceedings of the 40$^{\text{th}}$ Winter Conference on Simulation, Miami (FL), pp. 17–26.

[38] Jones SL, Sullivan AJ, Cheekoti N, Anderson MD & Malave D, 2004, *Traffic simulation software comparison study*, (Unpublished) Technical Report, Tuscaloosa (AL).

[39] Kidd M, 2016, Statistician in the Centre for Statistic Consultation, *Stellenbosch University*, [Personal Communication], Contactable at mkidd@sun.ac.za.

[40] Kleijnen JP, 1995, *Verification and validation on simulation models*, European Journal of Operational Research, **82(1)**, pp. 145–162.

[41] Klein LA & Kelley MR, 1996, *Detection technology for IVHS. Volume I: Final report*, (Unpublished) Technical Report, McLean (VA).

[42] Koonce P, Rodegerdts L, Lee K, Quayle S, Beaird S, Braud C, Bonneson J, Tarnoff P & Urbanik T, 2008, *Traffic signal timing manual*, (Unpublished) Technical Report, Transportation and Research Board, Portland (OR).

[43] Kotusevski G & Hawick K, 2009, *A review of traffic simulation software*, Research Letters in the Information and Mathematical Sciences, **13**, pp. 35–54.

[44] Krajzewicz D, Hertkorn G, Rössel C & Wagner P, 2002, *SUMO (Simulation of Urban MObility) — An open-source traffic simulation*, Proceedings of the 4$^{\text{th}}$ Middle East Symposium on Simulation and Modelling, Amman, pp. 183–187.

[45] Lämmer S, Donner R & Helbing D, 2008, *Anticipative control of switched queueing systems*, The European Physical Journal, Part B, **63(3)**, pp. 341–347.

[46] Lämmer S & Helbing D, 2008, *Self-control of traffic lights and vehicle flows in urban road networks*, Journal of Statistical Mechanics: Theory and Experiment, **2008(4)**, pp. 1–30.

[47] Lämmer S & Helbing D, 2010, *Self-stabilizing decentralized signal control of realistic, saturated network traffic*, (Unpublished) Technical Report, Santa Fe Institute, Santa Fe (NM).

[48] Law AM, 2008, *How to build valid and credible simulation models*, Proceedings of the 40$^{\text{th}}$ Winter Conference on Simulation, Miami (FL), pp. 39–47.

[49] Law AM & Kelton WD, 1991, *Simulation modeling and analysis, 1991*, 2$^{\text{nd}}$ Edition, McGraw-Hill Inc, New York (NY).

[50] Law A & Kelton D, 2000, *Simulation modelling and analysis*, 3$^{\text{rd}}$ Edition, McGraw-Hill, Boston (MA).

[51] Litman T, 2006, *London congestion pricing: Implications for other cities*, (Unpublished) Technical Report, Victoria Transport Policy Institute, Victoria (BC).

[52] Lowrie P, 1982, *The Sydney coordinated adaptive traffic system — Principles, methodology, algorithms*, Proceedings of the 207th International Conference on Road Traffic Signalling, London, pp. 67–70.

[53] Maria A, 1997, *Introduction to modeling and simulation*, Proceedings of the 29th Winter Conference on Simulation, Atlanta (GA), pp. 7–13.

[54] Martis MS, 2006, *Validation of simulation based models: A theoretical outlook*, The Electronic Journal of Business Research Methods, **4(1)**, pp. 39–46.

[55] McTrans, 2016, *TSIS-CORSIM*, [Online], [Cited October 18th, 2016], Available from `http://mctrans.ce.ufl.edu/mct/index.php/tsis-corsim/`.

[56] Mishan EJ & Mishan EJ, 1969, *The costs of economic growth*, The Journal of Finance, **24**, pp. 561–563.

[57] Odell J, 2002, *Agents and complex systems*, Journal of Object Technology, **1(2)**, pp. 35–45.

[58] Oh S, Ritchie SG & Oh C, 2002, *Real-time traffic measurement from single loop inductive signatures*, Journal of the Transportation Research Board, **1804(1)**, pp. 98–106.

[59] Osipenko L & Farr J, 2004, *System dynamics and dynamic systems integration in regulatory environments*, Proceedings of the 22th System Dynamics Society Conference, England, No page numbers.

[60] Placzek B, 2014, *A self-organizing system for urban traffic control based on predictive interval microscopic model*, Engineering Applications of Artificial Intelligence, **34**, pp. 75–84.

[61] Prevedouros PD & Li H, 2000, *Comparison of freeway simulation with INTEGRATION, KRONOS, and KWaves*, Proceedings of the 4th International Symposium on Highway Capacity, Maui (HI), pp. 96–107.

[62] Rakha H, Hellinga B & Van Aerde M, 1996, *Systematic verification, validation and calibration of traffic simulation models*, Proceedings of the 75th Transportation Research Board Annual Meeting, Washington (DC).

[63] Ratrout NT & Rahman SM, 2009, *A comparative analysis of currently used microscopic and macroscopic traffic simulation software*, The Arabian Journal for Science and Engineering, **34(1B)**, pp. 121–133.

[64] Sargent RG, 2003, *Verification and validation of simulation models*, Proceedings of the 37th Winter Conference of Simulation, Orlando (FL), pp. 130–143.

[65] Savrasovs M, 2011, *Urban transport corridor mesoscopic simulation*, Proceedings of the 25th European Council for Modelling and Simulation, Krakow, pp. 587–593.

[66] Sbayti H & Mahmassani H, 2006, *Optimal scheduling of evacuation operations*, Journal of the Transportation Research Board, **1964(1)**, pp. 238–246.

[67] Schultz BB, 1985, *Levene's test for relative variation*, Systematic Biology, **34(4)**, pp. 449–456.

[68] Seila AF, 1995, *Introduction to simulation*, Proceedings of the 27th Winter Conference on Simulation, Arlington (VA), pp. 7–15.

[69] Serugendo GDM, Gleizes MP & Karageorgos A, 2006, *Self-organisation and emergence in MAS: An overview*, Informatica, **30(1)**, pp. 45–54.

[70] Shalizi CR, 2001, *Causal architecture, complexity and self-organization in the time series and cellular automata*, PhD thesis, University of Wisconsin, Madison (WI).

[71] Shannon RE, 1998, *Introduction to the art and science on simulation*, Proceedings of the 30th Winter Conference on Simulation, Washington (DC), pp. 7–14.

[72] Software-informer, 2016, *CONTRAM*, [Online], [Cited October 18th, 2016], Available from http://contram.software.informer.com/.

[73] Solutions V, 2016, *VisSim*, [Online], [Cited October 18th, 2016], Available from http://www.vissim.com/company/about_visual_solutions.html.

[74] Sterman JD, 2000, *Business dynamics: Systems thinking and modeling for a complex world*, Irwin/McGraw-Hill, Boston (MA).

[75] *System Dynamic Simulation*, 2016, [Online], [Cited April 7th, 2016], Available from http://www.anylogic.com/system-dynamics.

[76] Systems IT, 2016, *SUMO — Simulation of Urban Mobility*, [Online], [Cited October 17th, 2016], Available from http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/.

[77] Szklarski J, 2010, *Cellular automata model of self-organizing traffic control in urban networks*, Bulletin of the Polish Academy of Sciences: Technical Sciences, **58(3)**, pp. 435–441.

[78] Taylor BD, 2002, *Rethinking traffic congestion*, Access Magazine, **1(21)**, pp. 8–16.

[79] Trafficware, 2016, *Synchro studio: Planning and analysis software*, [Online], [Cited October 16th, 2016], Available from http://www.trafficware.com/.

[80] Van Aerde M, Hellinga B, Baker M & Rakha H, 1996, *INTEGRATION: An overview of traffic simulation features*, [Online], [Cited September 2016], Available from https://pdfs.semanticscholar.org/5ba0/8ab26cdaa90ad0eb7c37a71b7146622096c6.pdf.

[81] Van der Merwe E, 2009, *Improving the flow of an isolated traffic system*, MA thesis, Stellenbosch University, Stellenbosch.

[82] Wavetronix, 2016, *Wavetronix SmartSensor Advance*, [Online], [Cited March 23rd, 2016], Available from http://www.wavetronix.com/en/products/smartsensor/advance.

[83] Webster FV, 1958, *Traffic signal settings*, (Unpublished) Technical Report, Road Research Laboratory, London.

[84] Wikipedia, 2016, *China National Highway 110 traffic jam — Wikipedia, The Free Encyclopedia*, [Online], [Cited November 15th, 2016], Available from https://en.wikipedia.org/w/index.php?title=China_National_Highway_110_traffic_jam&oldid=739214366.

[85] Williams LJ & Abdi H, 2010, *Fisher's least significant difference (LSD) test*, Encyclopedia of Research Design, pp. 1–5.

[86] Winston WL, 2004, *Operations research: Applications and algorithms*, 4th Edition, Cengage Learning, Belmont (CA).

[87] Xie X, Barlow GJ, Smith SF & Rubinstein ZB, 2011, *Selfscheduling agents for real-time traffic signal control*, (Unpublished) Technical Report, TR-RI-11-06, Robotics Institute, Carnegie Mellon University, Pittsburgh (PA).